Missouri S&T Library and Learning Resources

Scholars' Mine

Masters Theses

Student Theses and Dissertations

Spring 2014

Fair and optimal resource allocation in wireless sensor networks

Vinodhini Ravikumar

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses

 Part of the Electrical and Computer Engineering Commons

Department:

Recommended Citation

Ravikumar, Vinodhini, "Fair and optimal resource allocation in wireless sensor networks" (2014). *Masters Theses*. 7271.
https://scholarsmine.mst.edu/masters_theses/7271

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

FAIR AND OPTIMAL RESOURCE ALLOCATION IN WIRELESS SENSOR NETWORKS

by

VINODHINI RAVIKUMAR

A THESIS

Presented to the Faculty of the Graduate School of

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2014

Approved by

Dr. Maciej Zawodniok, Advisor
Dr. Jun Fan
Dr. Jagannathan Sarangapani

ABSTRACT

There is a large amount of research in wireless networks focuses on optimization of either network routing and power control alone. In contrast, this work aims at jointly optimizing the transmission power and routing path selection in order to optimize allocation of resources in interference constrained wireless environment. Moreover, we consider a multipath routing where multiple alternative paths are employed to transmit data between the end nodes. One of modern communication techniques that it applies to a network coding, though not explicitly implemented in this work. The proposed approach is first analyzed theoretically using Lagrangian optimization for a three-node scenario. We analyze this basic scenario, as it is essential for development of the overall multi-path routing schemes for multi-hop networks. The optimal solution for the three-node topology is replicated throughout the network to converge to a network-level solution. In contrast to existing studies, we explicitly consider interference from adjacent links, which varies with traffic flow thus optimizing the routing, and flow control decisions. The results and conclusions provide guidance as to the optimum routing decisions and a corresponding theoretical performance limits. The optimization of the throughput of the wireless network scenario is considered as a multi-variable optimization problem subject to flow and power constraints. Numerical analysis performed in Matlab-Simulink indicates that, given loose outage constraints, an optimal trade-off between the channel parameters renders optimum results even when the gain of the channel varies with time. The theoretical analysis and simulations demonstrate and validate that the channel capacity and efficiency are maximized when the routing decisions consider the network performance trade-offs.

Next, the proposed routing and power control scheme is experimentally evaluated in hardware using universal software radio peripheral (USRP2). The USRP testbed utilizes the proposed multi-variable optimization algorithm. The communication system is implemented using GNU Radio software where the physical layer employs two direct-spread spectrum variants: (a) binary phase shift keying (DS-BPSK) and (b) orthogonal frequency division modulation (DS-OFDM) schemes. The experimental results are compared with the simulation results.

ACKNOWLEDGMENTS

At the outset, I would like to thank God Almighty for bestowing the blessings upon me all along. It is a pleasure to extend my warm thanks to all those kind people whose support and help has enabled the successful completion of this thesis.

Foremost, I would like to express my deep sense of gratitude to Dr. Maciej J. Zawodniok, my advisor and a source of inspiration. His enthusiastic encouragement, guidance and support was a great motivation towards my work throughout my tenure at this university. I would also like to acknowledge National Science Foundation for supporting the research through Grant.

I am grateful to my committee members, Dr. Jagannathan Sarangapani and Dr. Jun Fan for their kind support. I was privileged to have an enriching learning experience through the fruitful interactions, both inside and outside the classroom. I would like to thank the faculty and staff of the ECE Department of Missouri S&T who have helped me in many ways throughout my graduate study years.

I would like to extend my heartfelt thanks to all my research group members. I would indeed rejoice the wonderful moments and fruitful discussions we had, technical and otherwise while at work in ERL.

I am fortunate to have an extended family of friends at my home away from home. I would like to express many thanks to all of them and my entire network of close friends in Rolla who made my stay here a pleasant one. I also wish to extend my great thanks to all my dear friends from college and high school, who have rendered immense help and support throughout.

I am beholden to my entire family for providing a loving environment and kind support all along. Lastly and most importantly, words cannot express how indebted I am to my wonderful parents, Mr. Ravikumar Krishnaswamy, Mrs. Radhika Ravikumar and my younger brother Aravind Ravikumar for their unconditional support and love, which has been the most significant factor all my life and also through my graduate work. To them I dedicate this thesis.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The traditional routing schemes, including the multipath ones, assume fixed maximum capacity on communication links. The link's performance is assumed to vary mainly due to both total traffic passing this link and the protocol stack efficiency. However, in wireless networks the capacity of a particular link changes with traffic and interference from neighbor links. Consequently, when a routing decision is made to forward packets to one of the intermediate nodes, the capacity of adjacent wireless links will be reduced. Such inter-link interference can easily hinder the performance and optimality of any routing decision or scheme. Moreover, the fairness in terms of amount of capacity allocated to each flow-source will be significantly altered by such suboptimal routing decision. Hence, the ideal scheme should take into account not only current capacities of the alternative paths but also the effect of the routing decision on the overall network capacity and performance. The proposed methodology derives the optimal forwarding strategy in a general multi-path routing scenario in wireless sensor networks. However, the derived solution can be extended to more general ad hoc and peer-to-peer networks, which will be undertaken as a future work.

The recent advances in sensor and chip manufacturing technology have resulted in the ability to integrate sensors, radio communications, and micro-controller into a single integrated circuit (IC) package. This capability is enabling networks of very low cost sensors that are able to communicate with each other using low power wireless data routing protocols. In a typical wireless sensor network (WSN), the data collected, processed at the sensors are relayed to base station, also called a gateway or access point. In a large sensor networks the base station might be far away from the sensor thus requiring intermediate sensor node to relay the information in a multi-hop fashion. However, the existing routing protocols for WSNs tend to focus on low power, low duty cycle operation that has low network throughput requirements. In contrast, more and more contemporary sensing applications generate high cumulative, traffic from large number of sources in the network. Hence, it becomes important to emphasize the communication performance while maintaining fairness among the sensors data flows. Consequently, the routing decisions have to guarantee

optimization of network performance. This work aims at deriving the optimal routing decisions based on theoretical capacity limits for various options.

The problem of optimal resource allocation in communication networks has been dealt with using a variety of approaches. The first approach dealing with *Network Utility Maximization* has been given by Kelly et al. in their seminal paper [3]. The NUM framework deals with the problem of optimal network resource allocation and solves it using a set of differential equations. That paper produced results for elastic type of traffic that described wired network scenario. But, practical wireless network has mostly inelastic form of traffic that varies significantly from TCP-like traffic.

Mobile wireless sensor networks, as shown in Fig. 1.1, have many advantages [11] over the conventional static network. There is a common misconception that the static network is more secure because of its known location, which can be protected. A mobile node's position is hard to mimic and hence when two nodes interact in a password protected environment, mobile nodes have far better security than the normal static ones [6]. The other advantages of mobile nodes include the amount of energy that can be conserved in various scenarios. When the nodes appear to be stationary, it is easier to conserve the energy by optimal usage. However, a proper routing scheme has potential to conserving energy also in the case of mobile nodes [5] . The mobile WSNs should have a highly cooperative communication channel in terms of independence and also sharing network protocols. The mobile sensor networks require that hierarchy be established between various lower-level nodes and nodes which process high-level data and information.

Typically, wireless applications have to operate with an allocated amount of battery power. Their battery supply is very minimal as compared to other static nodes and hence must learn to cope up with small ratio of power for longer periods [7] . Hence, while considering the performance metric for the operation of a wireless network, it is essential that energy consumption be an important part of the analysis section. The radio transceiver still consumes a lot of power in terms of receiving and transmitting nodes.

The presented preliminary work considers the multi-path routing decision for relay node. Two cases are presented in this work including single traffic flow passing through a node where alternative paths are available, and a case where multiple traffic

flows merge in such topology. Both the power control and traffic split is controlled by the derived solution. The power control aims at achieving the optimal link capacity to at lowest transmission energy cost. It considers interference from other nodes in the neighborhood and traffic split among the alternative paths. In concert, the traffic flow is split among the available alternative paths such that the overall, end-to-end throughput is increased while minimizing energy consumption. Further extension of the work will generalize the presented methodology to an arbitrary large topology and number of alternative routing paths. The preliminary work demonstrates the overall approach in simplified scenarios, while the more complex scenarios can be reduced to such simple problems using divide-and-conquer approach.

The main contributions of the work include: (1) the theoretical derivation of an optimal routing strategy, (2) analysis of its energy-efficiency and optimality, (3) the study of the trade-off between channel capacity and the power requirements, (4) simulation evaluation of the proposed multi-path, multi-hop routing scheme, and (5) and experimental validation of the proposed multi-path routing with power control.

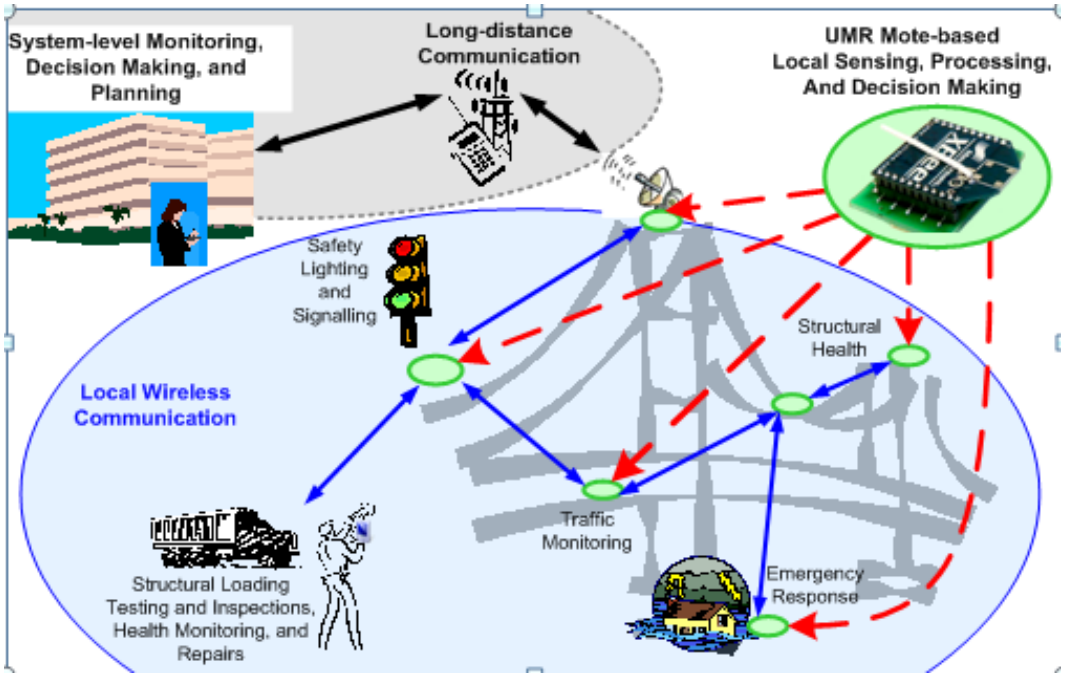

Figure 1.1. Mobile Wireless Sensor Network Framework

The rest of the paper is organized as follows. The literature review and Related Work is presented in the Section 1 and Section 2 respectively. The system model and the Lagrangian method based objective function for optimal routing strategy is explained in Section 3. Then, the optimal solution is derived. The experimental design using software defined radios is presented in Section 4. The Section 5 describes a trade-off between power and capacity and gives an analysis of the simulation results. Later, Section 6 offers a perspective on the experimental results. Finally, Section 7 offers conclusions and future works.

2. RELATED WORK

In the past, mobile WSNs have been analyzed in different perspectives. A sensor node in a multi-hop network has to transmit both its own traffic and forward other node's data. Hence, besides the competition with nodes for transmitting to the same base station, there is contention in transmitting its own information along with the relayed node's information. This type of a problem is practically not a part of fixed wireless networks, where there is a single connection between the user node and the base station. Many researchers studied an issue of routing packets over long distances using multi-hop approach. Their results vary upon the initial conditions taken by them and also their method of approach [11]. The control of trade-offs between various parameters has also been discussed using concept of adaptive resource allocation.

Power consumption by networks that transmit information over longer hops has been studied and efficiency in generally following lesser number of hops with lesser delay. The same concept has been discussed from the point of view of reception cost and also totally energy consumed in these cases.

This paper evaluates theoretical capacity limits assuming the underlying routing scheme while maintaining fairness in sharing resources among the data flows. The maximum end-to-end throughput or capacity between a given source and destination pair are derived in the presence of interference from other nodes. This analysis is done for two different topologies. The signal to noise ratio is a basic parameter that affects the fairness in a network environment since it determines the available capacity.

This particular analysis is done for a three-node case. In the first variant of this case, the node 1 transmits information to node 2 via node 3 as the relay node. In the second variant of this case, the node 1 transmits information to node 2 via node 3 and node 3 simultaneously transmits it's own information to node 2. Node 2 acts as the destination node in both the variations. By using the method of Lagrangian multipliers, multi-variable optimization is done for transmit power values used by each node; taking into consideration the maximum end-throughput that can be achieved.

For a dynamically routed traffic, it is desirable for packets with same source and destination to follow different paths. While taking multiple paths to the destination, the path taken becomes dependent on the relative load on candidate paths at flow-arrival time. This helps in flows through congested links. By load balancing, improved response time can be achieved. In our case, after the optimization, the packets can take the route that is more feasible to them in terms of net throughput. The existing methods route the packet based on a fixed topology and load balancing is done based on that. In our optimization method, we try to create a dynamic environment by frequent changing of the node positions and randomizing the gain or attenuation of the channel.

The proposed method derives the optimal solution algebraically. This gives a platform to theoretically test out the various possible solutions for fair throughput allocation. Also, by considering the mutual interference from the nearby nodes, we create a more realistic scenario and are able to account for all kinds of disturbances in the final optimal solution. This gives us a more accurate solution for power split and hence optimal throughput.

3. OVERVIEW OF THE PROPOSED MULTI-VARIABLE OPTIMIZATION

We consider a routing decision for three nodes scenario which when repeated at each hop will render the network-wide routing scheme. The throughput of the wireless network is said to be optimal when the optimal power values for all links is achieved. The traffic flow splitting among the multiple paths is directly derived from the resulting link capacities. This is a multi-variable optimization problem where the channel gain or attenuation is a time-varying quantity. The optimization problem is formulated using the Lagrangian multipliers method. It yields the maxima or minima of a given objective function subject to a given set of constraints. The general formulation of the method is as follows:

$$maximize\ f\ (x,y), \quad subject\ to\ g\ (x,y) = \ c \tag{1}$$

where, the f(x,y) is the given function and the constraints are given by g(x,y)=c.

Algorithm 1 End-to-end throughput maximization algorithm

```
Initialize the variables and formulate the objective function
while
Current Node is not equal to destination do
  for
  Each 3 hop network from 1 through M do
    Set Power constraint for each link as a function of incoming flow
    Formulate the Lagrangian multiplier function
    Solve for optimal cases of shadow price lambda and power combination
    if
    feasible solution satisfying constraints then
      Compute Net throughput of 3 links for different cases
      Compare net throughput of different cases to obtain maximum
    end if
  end for
end while
Compute end-to-end throughout for the entire network
```

We, then formulate a Lagrangian function using a new variable called Lagrange multiplier λ. The Lagrangian function is defined as follows:

$$L\ (x, y, \lambda) = f\ (x, y)\ + \lambda(g\ (x, y) - c) \tag{2}$$

If f (x,y) is a maximum for the constrained optimization problem, then there exists a λ such that (x,y,λ) is a stationary point for the Lagrangian function. Not all the stationary points lead to a proper maximum. Thus the method of Lagrange multipliers gives a necessary condition for the solution to be optimum in a constrained scenario.

The method of Lagrange multipliers deals only with equality constraints and the problem case presented here represents inequality constraints. The Karush-Kuhn Tucker conditions (K.K.T conditions)[**?**] allow for inequality conditions to be present, thereby generalizing the method of Lagrange multipliers for all kinds of optimization possible related to non-linear programming.

Therefore, the K.K.T conditions are the necessary conditions for some feasible x to be minimum for the optimization problem mentioned above equation.

Next, we maximize $f(x, y)$ with respect to $s_i(x, y) \leq 0$, where $i = 0, 1, ...h + 1$ and $s_i(x, y) = 0$ where $i = h + 1, ..., m$ (h and m are integers) represents the K.K.T conditions.

3.1. FIRST TRIPLET SUB-NETWORK

Optimization of throughput of a wireless network is a result of fair sharing of the network resources. This is a multi-variable optimization problem as the power transmitted by each node as well as traffic parameters of each link becomes variable. Each link is analyzed individually and the network flow as a whole and optimization is performed. The Fig. 3.1 shows the topology we are going to analyze. The information flow h enters through node 1 and gets relayed through the rest of the network in a fair and efficient manner. We begin our analysis of optimal power allocation by taking the first part of the network. This is given by the nodes 1-2-3 combination.

In the topology shown in Fig. 3.2, we assume unidirectional communication from nodes (1,2) , (1,3) and bi-directional communication between (2,3). This assumption helps us in extending this version of the topology to a larger-scale version.

We begin by defining the capacity flow through each link for the 3-node case. Assuming the theoretical capacity limit to be given by the Shannon's Capacity Formula, We get,

$$X_{12} = B \ \log_2 \left(1 + \frac{P_{12}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) \tag{3}$$

$$X_{13} = B \ \log_2 \left(1 + \frac{P_{13}g_{13}}{P_{12}g_{13} + P_{23}g_{23} + \eta_3}\right) \tag{4}$$

$$X_{32} = B \ \log_2 \left(1 + \frac{P_{32}g_{32}}{P_{12}g_{12} + P_{13}g_{12} + \eta_2}\right) \tag{5}$$

$$X_{23} = B \ \log_2 \left(1 + \frac{P_{23}g_{23}}{P_{12}g_{13} + P_{13}g_{13} + \eta_3}\right) \tag{6}$$

The optimization problem in Wireless sensor networks arises as a result of a number of decisions available for routing information through the nodes. This leads to a net throughput, which may or may not be the optimum. So, fairness in throughput arises by proper allocation of the network resources. Here, the multi–variable optimization problem is formulated as given below.

According to the flow conservation law,

$$\begin{aligned} X_{13} + X_{12} &= h \\ X_{12} + X_{23} &= nh + X_{32} \\ X_{12} + X_{32} &= (1-n)h + X_{23} \end{aligned} \tag{7}$$

Here, we consider both nodes 2 and 3 to be the forwarding nodes. So, based on the optimal power allocation, nh and (1-n)h flows get routed.

Other constraints include power transmitted by each node. They are as follows:

$$\begin{aligned} P_{12} &\leq P_{max} \\ P_{13} &\leq P_{max} \\ P_{32} &\leq P_{max} \\ P_{23} &\leq P_{max} \end{aligned} \tag{8}$$

Using these flow constraints and the power constraints as the base for multi-variable optimization, we use the Lagrange multiplier method. Lagrange Multiplier method works on the assumption that the objective function is a concave function to be maximized and the constraint belongs to a convex set.

The objective function for optimization of this three-node case is

$$F = X_{13} + X_{12} \tag{9}$$

$$= h \tag{10}$$

The Lagrange equation is given as:

$$\begin{aligned} L =& X_{12} + X_{13} - \lambda_1(P_{12} - P_{max}) - \lambda_2(P_{13} - P_{max}) - \lambda_3(P_{32} - P_{max}) - \\ & \lambda_4(P_{23} - P_{max}) \end{aligned} \tag{11}$$

$$\begin{aligned} L =& B\ \log_2\left(1 + \frac{P_{12}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) + B\ \log_2\left(1 + \frac{P_{32}g_{32}}{P_{12}g_{12} + P_{13}g_{12} + \eta_2}\right) \\ & - \lambda_1(P_{12} - P_{max}) - \lambda_2(P_{13} - P_{max}) - \lambda_3(P_{32} - P_{max}) - \lambda_4(P_{23} - P_{max}) \end{aligned} \tag{12}$$

The above equation forms the base for Lagrange multiplier method of multi-variable optimization. By Using, Karush Kuhn-Tucker Conditions, we say that if the objective function and the constraints are continuously differentiable, then by partial differentiation of the Lagrangian multiplier equation with respect to each variable and each λ, we get the optimal solution for power allocation.

λ is known as the shadow price. The Shadow price defines the change in the objective function with respect to that particular constraint equation.

Now, the partial differentiation of the Lagrange equation with respect to the independent variables P_{12}, P_{13}, P_{32}, P_{23}, λ_1, λ_2, λ_3 and λ_4. Here,λ denotes the Lagrangian multiplier or the shadow price.

Now, differentiating L with respect to P_{12} we get:

$$\begin{aligned} L_{12} =& B * 1.442 \ \log_2 \left(1 + \frac{P_{12}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) \left(\frac{\dot{g_{12}}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) \\ & - B * 1.442 \ \log_2 \left(1 + \frac{P_{32}g_{32}}{P_{12}g_{12} + P_{13}g_{12} + \eta_2}\right) \left(\frac{P_{32}\dot{g_{32}}g_{12}}{P_{13}g_{12} + P_{12}g_{12} + \eta_2}\right)^2 - \lambda_1 \end{aligned} \tag{13}$$

Similarly, differentiating L with respect to P_{13} , P_{32} and P_{23} we get:

$$\begin{aligned} L_{13} =& B * 1.442 \ \log_2 \left(1 + \frac{P_{13}g_{13}}{P_{12}g_{13} + P_{23}g_{23} + \eta_3}\right) \left(\frac{\dot{g_{13}}}{P_{12}g_{13} + P_{23}g_{23} + \eta_3}\right) - \\ & B * 1.442 \ \log_2 \left(1 + \frac{P_{12}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) \left(\frac{P_{12}\dot{g_{12}}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right)^2 - \lambda_2 \end{aligned} \tag{14}$$

$$L_{32} = -B * 1.442 \ \log_2 \left(1 + \frac{P_{12}g_{12}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right) \left(\frac{P_{12}\dot{g_{12}}g_{32}}{P_{13}g_{12} + P_{32}g_{32} + \eta_2}\right)^2 - \lambda_3 \tag{15}$$

$$L_{23} = -B * 1.442 \ \log_2 \left(1 + \frac{P_{13}g_{13}}{P_{12}g_{13} + P_{23}g_{23} + \eta_3}\right) \left(\frac{P_{13}\dot{g_{13}}g_{23}}{P_{12}g_{13} + P_{23}g_{23} + \eta_3}\right)^2 - \lambda_4 \tag{16}$$

Also, differentiating L with respect to λ_1, λ_2, λ_3 and λ_4, we get :

$$\lambda_1(P_{12} - P_{max}) = 0 \tag{17}$$

$$\lambda_2(P_{13} - P_{max}) = 0 \tag{18}$$

$$\lambda_3(P_{32} - P_{max}) = 0 \tag{19}$$

$$\lambda_4(P_{23} - P_{max}) = 0 \tag{20}$$

So, now by solving the Lagrangian equations and the equations involving λ simultaneously, we obtain 5 cases of optimality:

$$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0, \lambda_4 = 0 \tag{21}$$

$$\lambda_1 = 0, \lambda_2 = 0, P_{32} = P_{max}, P_{23} = P_{max} \tag{22}$$

$$\lambda_1 = 0, P_{13} = P_{max}, P_{32} = P_{max}, P_{23} = P_{max} \tag{23}$$

$$P_{12} = P_{max}, \lambda_2 = 0, P_{32} = P_{max}, P_{23} = P_{max} \tag{24}$$

$$P_{12} = P_{max}, P_{13} = P_{max}, P_{32} = P_{max}, P_{23} = P_{max} \tag{25}$$

We also see that, both P_{32} and $P_{23} = P_{max}$. This is obtained by differentiating L with respect to P_{32} and P_{23} and seeing that they can only equal P_{max}.

Also, we obtain two distinct equations when λ_1 and λ_2 are equated to zero. They are:

$$\begin{aligned} &P_{12}^2(g_{12}g_{13}^2) + P_{12}((2g_{13}g_{12}(P_{23}g_{23} + \eta_3))) + ((g_{12}(P_{23}g_{23} + \eta_3)^2) + (g_{12}P_{13} \\ &g_{13}(P_{23}g_{23} + \eta_3)) - (P_{13}g_{13}^2(P_{13}g_{12} + P_{32}g_{32} + \eta_2))) = 0 \end{aligned} \tag{26}$$

$$\begin{aligned} &P_{13}^2(g_{13}g_{12}^2) + P_{13}((2g_{13}g_{12}(P_{32}g_{32} + \eta_2))) + ((g_{13}(P_{32}g_{32} + \eta_2)^2) + (g_{12}P_{12} \\ &g_{13}(P_{32}g_{32} + \eta_2)) - (P_{12}g_{12}^2(P_{12}g_{13} + P_{23}g_{23} + \eta_3))) = 0 \end{aligned} \tag{27}$$

By solving the above the above equations for the five specific cases, we obtain 5 optimal solution sets for power transmitted. Our main aim here is to find the best optimal throughput for the entire network. This can be found by using these power values in the Shannon's capacity equation. This analysis is completed in the next section by simulation and further tabulation. The simulation is performed for one of the five optimal cases for simplification purposes and due to page constraints.

Theorem 1. (Case with First-Three-Hop-Nodes): *The Optimal Solution for Maximizing the end-to-end throughput for a three node case has at least one link operating at* P_{max}.

$$P_{ij} = \frac{-b_{ij} \pm \sqrt{b_{ij}^2 - 4a_{ij}c_{ij}}}{2a_{ij}} \tag{28}$$

where, in our scenario, i=1, j=(2,3) and a_{ij} *is the coefficient of the square term,* b_{ij} *is the coefficient of the unit term and* c_{ij} *is the constant term.*

The traffic split factor r *is decided based on the value of the Capacity* C_{32} *if,*

$$\begin{aligned} if \quad & C_{32} + C_{13} < C_{12} \quad then \ 0 < r < 0.5 \\ elseif \quad & C_{32} + C_{13} > C_{12} \quad then \ 0.5 < r < 1 \\ & else \quad r = 0.5 \end{aligned} \tag{29}$$

This algorithm flow decides the gradual traffic splitting in our network topology. The proof for this theorem is fairly explicit from the equations 18 *to* 22*. As per our assumptions, by linking* λ_3 *with respect to* P_{32} *and* λ_4 *with respect to* P_{23}*, we get* P_{32} *and* P_{23} *as being constant at* P_{max}*.*

By using different combinations of the same objective function, we still end up with one of the link powers in the three node case as always being P_{max}*. This validates our lemma one.*

Lemma 2. *The Optimal end to end throughput by fair allocation of power for three interfering links is a non binary set given by different combinations of equation* 23*,* 24 *,* P_{32} *equal to* P_{max} *and* P_{23} *equal to* P_{max}*.*

3.2. SUBSEQUENT TRIPLET SUB-NETWORKS

Now, to extend this particular three-node scenario to a larger topology, consider the next three nodes in the Fig. 3.3. We try applying the same three-node principle to these nodes and optimize the end-to-end throughput.

We start from the highlighted section 3-6-4. So, applying the same principle as in Lemma 1, that is having one set of links operating at P_{max}, we try finding the optimum operating conditions for the other two links.

In the previous 3-hop optimization, we assumed the maximum power on each link to be equal to P_{max} for a flow of h units. But, the maximum power for this next 3-hop depends on the incoming flow, that is X_{13}+X_{23}-X_{32}.

Using, the capacity constraints on each link, that is, flow through each link does not exceed the maximum allowable capacity limits through the link. Here,

$$\begin{aligned} C_{36} &\leq C_{max} \\ C_{34} &\leq C_{max} \end{aligned} \tag{30}$$

From the above equations, we get, the maximum power constraints as,

$$\begin{aligned} P_{36} &\leq ((e^{X_{cum1}} - 1) * (\frac{I_6}{g_{36}}) \\ P_{34} &\leq ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{34}}) \end{aligned} \tag{31}$$

Also, the total interference at node 6 due to all other nodes including thermal noise is given by I_6 and that at node 4 is given by I_4,

$$\begin{aligned} I_6 &= P_{34}g_{36} + P_{1c}g_{16} + P_{2c}g_{26} + P_{56}g_{56} + \eta_6 \\ I_4 &= P_{36}g_{34} + P_{1c}g_{14} + P_{2c}g_{24} + P_{54}g_{54} + P_{64}g_{64} + \eta_4 \end{aligned} \tag{32}$$

where,

$$\begin{aligned} X_{cum1} &= X_{13} + X_{23} - X_{32} \\ P_{2c} &= P_{23} + P_{24} + P_{25} + P_{26} \\ P_{1c} &= P_{12} + P_{13} \end{aligned} \tag{33}$$

We assume the objective function here to be maximizing the incoming flow, that is, summation of X_{36} and X_{34} with respect to power constraints on each link and also based on the 3 node optimized values.

Proceeding similarly as in previous case, the objective function for optimization is equivalent to maximizing the part of the initial flow h through node 3.

The objective function for optimization of this three-node case is

$$F = X_{36} + X_{34} \tag{34}$$

$$= X_{13} + X_{23} - X_{32} \tag{35}$$

The Lagrange equation is given as:

$$\begin{aligned} L = {} & X_{36} + X_{34} - \lambda_1(P_{36} - ((e^{X_{cum1}} - 1) * (\frac{I_6}{g_{36}}))- \\ & \lambda_2(P_{34} - ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{34}})) \\ & = B \ \log_2\left(1 + \frac{P_{36}g_{36}}{I_6}\right) + B \ \log_2\left(1 + \frac{P_{34}g_{34}}{I_4}\right) - \\ & \lambda_1(P_{36} - ((e^{X_{cum1}} - 1) * (\frac{I_6}{g_{36}})) - \lambda_2(P_{34} - ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{34}})) \end{aligned} \tag{36}$$

Similarly, differentiating the Lagrangian multiplier equation with respect to the variables, we get λ_1 and λ_2 as,

$$\begin{aligned} \lambda_1 = {} & \frac{B * 1.442}{(e^{X_{cum1}} - 1)^2} \ (\left(\frac{g_{36}}{I_6 + P_{36}g_{36}}\right) - \left(\frac{P_{34}g_{34}^2}{I_4^2 + P_{34}g_{34}I_4}\right) + \left(\frac{g_{34}(e^{X_{cum1}} - 1)}{I_4 + P_{34}g_{34}}\right) \\ & - \left(\frac{P_{36}g_{36}^2(e^{X_{cum1}} - 1)}{I_6^2 + P_{36}g_{36}I_6}\right)) \end{aligned} \tag{37}$$

$$\lambda_2 = \frac{B * 1.442}{(e^{X_{cum1}} - 1)^2} \ (\left(\frac{g_{36}}{I_6 + P_{36}g_{36}}\right) - \left(\frac{P_{34}g_{34}^2}{I_4^2 + P_{34}g_{34}I_4}\right)) + \left(\frac{\lambda_1}{(e^{X_{cum1}} - 1)}\right) \tag{38}$$

So, now by proceeding in the same way as the previous three-node case, we obtain 4 cases of optimality. These four cases of optimality are linked to Lagrangian.

They are as follows:

$$\begin{aligned}
\lambda_1 = 0, \lambda_2 = 0, P_{64} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{64}}) \\
\lambda_1 = 0, P_{34} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{34}}), P_{64} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{64}}) \\
P_{36} = ((e^{X_{cum1}} - 1) * (\frac{I_6}{g_{36}}), \lambda_2 = 0, P_{64} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{64}}) \\
P_{36} = ((e^{X_{cum1}} - 1) * (\frac{I_6}{g_{36}}), P_{34} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{34}}), \\
P_{64} = ((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{64}})
\end{aligned} \tag{39}$$

We had already known from the previous optimization that P_{64} is equal to $((e^{X_{cum1}} - 1) * (\frac{I_4}{g_{64}})$. This gets applied here to this three-node case also. Also, we obtain two distinct equations when λ_1 and λ_2 are equated to zero. They are:

$$\begin{aligned}
&P_{36}^3(g_{36}^2 g_{34}^2(e^{X_{cum1}} - 1)) + P_{36}^2((g_{36}^2(I_4 - P_{36}g_{34})^2(e^{X_{cum1}} - 1)) + (g_{36}^2 g_{34}(I_4 - P_{36}g_{34}) \\
&(e^{X_{cum1}} - 1)) - (g_{36}I_6 g_{34}^2) - (g_{36}g_{34}^2 I_6)(e^{X_{cum1}} - 1)) - P_{36}((2g_{36}I_6 g_{34}(I_4 - P_{36}g_{34})) \\
&+ (g_{36}g_{34}(I_4 - P_{36}g_{34})I_6(e^{X_{cum1}} - 1)) + (g_{34}^2 I_6^2(e^{X_{cum1}} - 1))) - (((I_4 - P_{36}g_{34}) \\
&I_6^2 g_{34}(e^{X_{cum1}} - 1)) - (I_6^2 P_{34}g_{34}^2) + (P_{34}g_{36}g_{34}(I_4 - P_{36}g_{34})I_6)) = 0
\end{aligned} \tag{40}$$

$$\begin{aligned}
&P_{34}^3(zg_{36}^2 g_{34}^2) + P_{34}^2((2zg_{34}^2 g_{36}(I_6 - P_{34}g_{36})) - (zg_{36}^2 g_{34}I_4) + (z.g_{34}^2 g_{36}^2 P_{36}) + (g_{36}^2 g_{34} \\
&I_4)) + P_{34}((2I_4 g_{36}g_{34}(I_6 - P_{34}g_{36})) + (I_4 g_{34}g_{36}^2 P_{36}) + (z(I_6 - P_{34}g_{36})^2 g_{34}^2) + (zg_{34}^2 \\
&P_{36}g_{36}(I_6 - P_{34}g_{36})) - (zg_{36}^2 I_4^2) - (zg_{36}g_{34}I_4(I_6 - P_{34}g_{36})) - (P_{36}g_{34}g_{36}^2 I_4))+ \\
&(((I_6 - P_{34}g_{36})^2 I_4 g_{36}) + ((I_6 - P_{34}g_{36})P_{36}g_{36}g_{34}I_4) \\
&- (zg_{36}(I_6 - P_{34}g_{36})I_4^2) - (P_{36}g_{36}^2 I_4^2)) = 0
\end{aligned} \tag{41}$$

where,

$$z = (1 + (e^{X_{cum1}} - 1)^2) \tag{42}$$

The solution for the above two cubic equations is of the general form:

$$P = \left(q + (q^2 + (r - p^2)^3)^{0.5}\right)^{1/3} + \left(q - (q^2 + (r - p^2)^3)^{0.5}\right)^{1/3} + p \quad (43)$$

Here,

$$\begin{aligned} p &= \frac{-b}{3a} \\ q &= p^3 + \frac{bc - 3ad}{6a^2} \\ r &= \frac{c}{3a} \end{aligned} \quad (44)$$

where, a is the coefficient of the cubic term, b is the coefficient of the square term, c is the coefficient of the unit term and d is the constant term.

Now, following a similar approach to the triangular area 2-5-4, we try to achieve optimality by using the adjacent link optimized values as constraints. Recursively, the Theorem 1 and Lemma 1 are employed to achieve optimality in all scenarios.

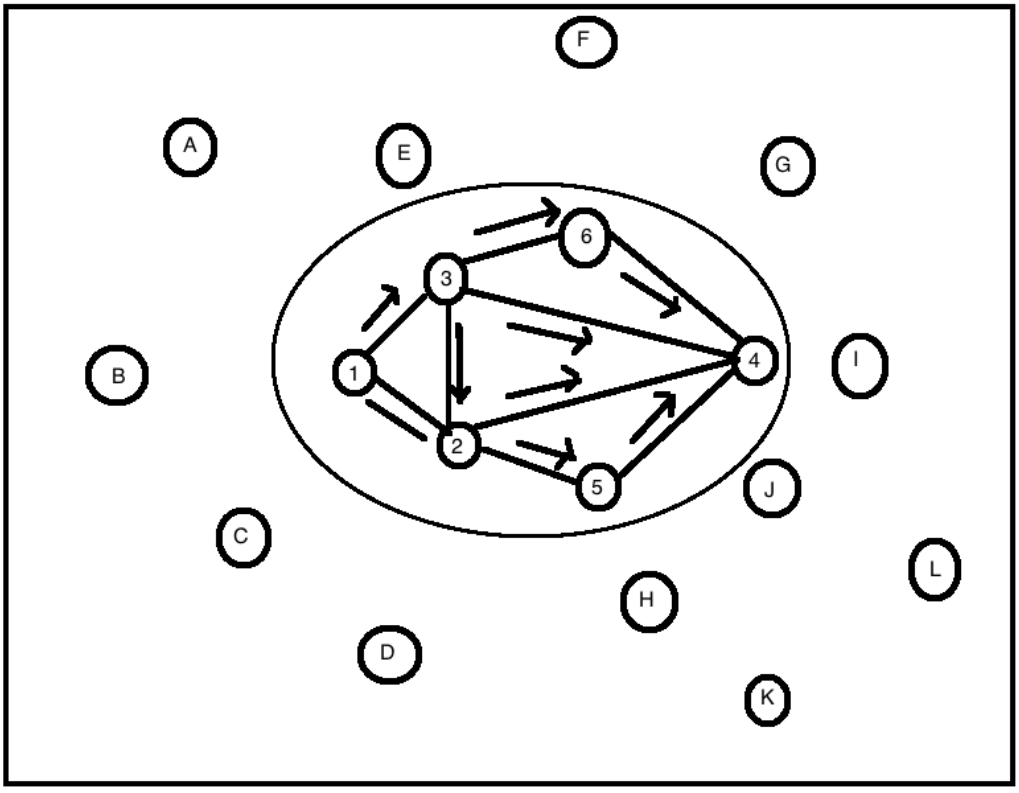

Figure 3.1. Basic Network Topology

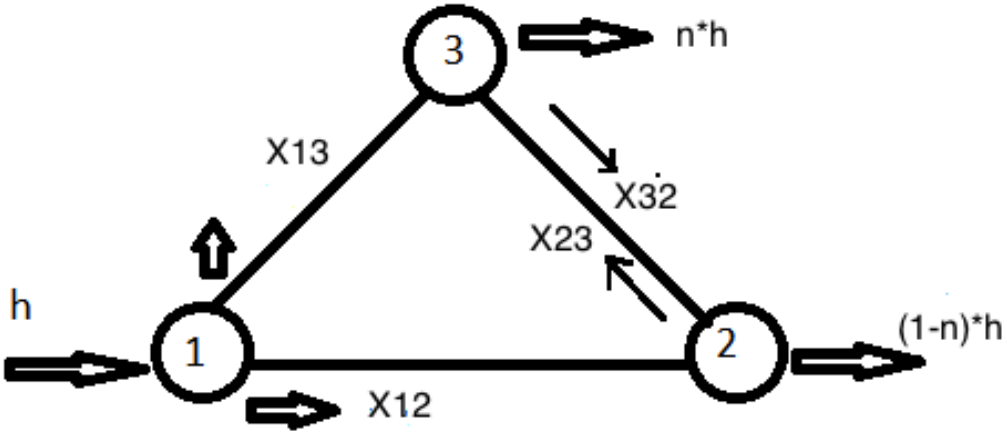

Figure 3.2. Network Topology 3-Node Case

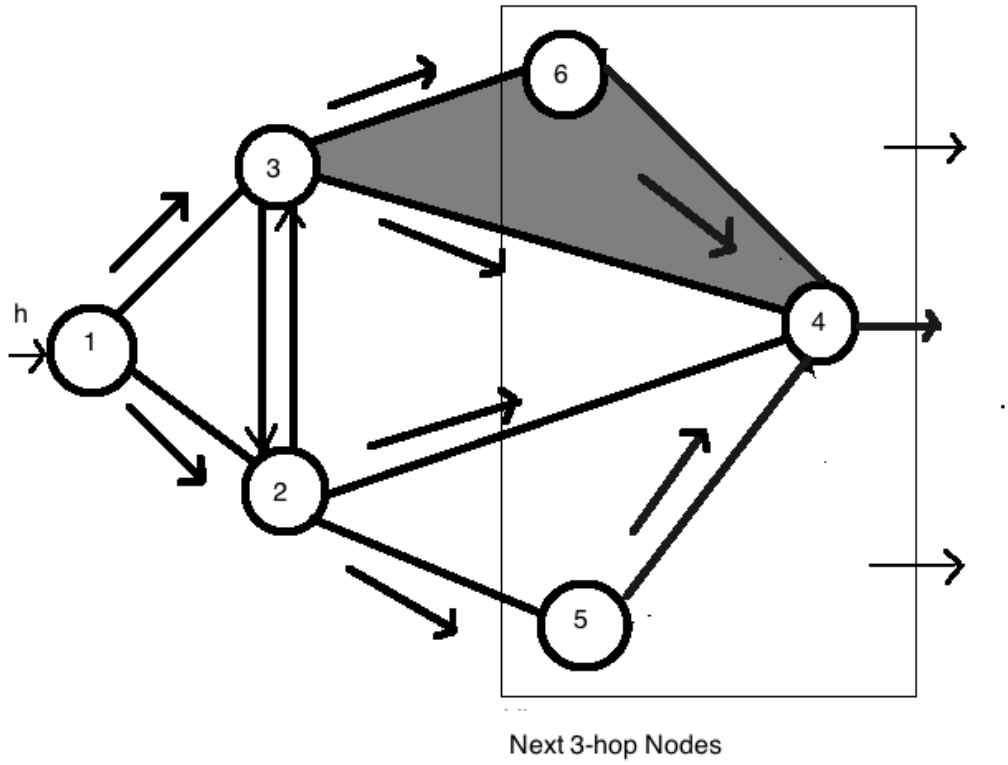

Figure 3.3. Network Topology Subsequent 3-Node Case

4. EXPERIMENTAL DESIGN USING SOFTWARE DEFINED RADIOS

Software defined radios (SDR) are used to emulate many radios working at different frequencies. It is used as a tool in experimenting with mobile and wireless communications. Initially in the nineties, this started as a trend using Digital signal processing as a tool and many of the radio functions were performed at satisfactory performance levels. This includes digital filtering performance improvement, noise reduction etc. SDR is characterized by increased flexibility and even slight modifications in the software programs can considerably change the functionality being tested [26, 27]. Hence, this satisfies the operating needs of individual applications. This helps in minimizing the hardware change for small changes in overall functionality.

Many of the mathematical functions can be incorporated into the hardware to perform digital processing of radio-frequency signals, for frequency selection and also for baseband signal conversion. In SDR, all the parameters used for experimentation are set in software and configurable through software. This makes the development and research process faster and cheaper by giving way to the rapid variations in configurable prototypes. The concept of Software defined radios was basically introduced in the 1970's [29]. It was initially used for defense communication by the U.S Department of Defense Office and was later adopted by commercial communication systems for research and development [28]. The conceptual diagram of SDR is shown in Fig. 4.1.

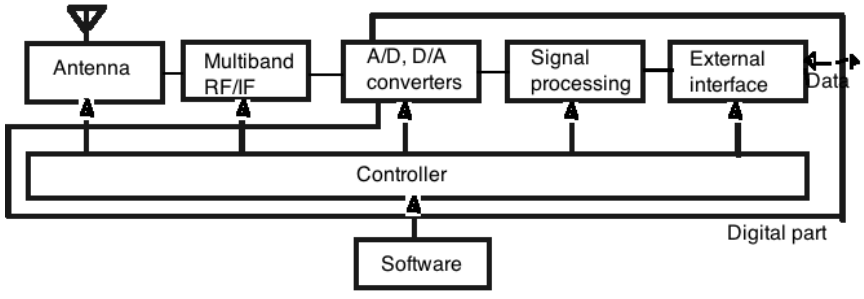

Figure 4.1. Conceptual Diagram of Software defined radio

4.0.1. Basic Principle of SDR. The basic principle of Software defined radios is minimizing the use of hardware and moving all possible processing to the software domain. The main difference between traditional radios and SDR's is that, traditional radios depend on dedicated hardware to perform their signal processing functions, but SDR's rely on software to configure their parameters. In its most basic form, the software defined radio as shown in Fig. 4.1, may consist of only an analog-to-digital convert chip connected to an antenna, with all the filtering, processing and detection taking place in the software.

Some of the hardware requirements for an SDR are, an ADC/DAC module, a multi-band RF/IF converter module to make the hardware work at desired frequency levels and a PC to do the software processing. Compared to traditional radios, the power-level requirement for a software radio is much high because of the complex processing taking place in the software, hence the need for powerful PCs. Depending upon the application being implemented, we might need narrow band or wide band processing [28]. For Narrow-band applications, a Pentium PC with reduced processing power is sufficient. For wide-band applications like OFDM or 8-PSK, we might require higher processing power to process information about independent channels.

In our application, we will be implementing a communication system based on DS-DBPSK and DS-OFDM for a wide-band application. The ideal SDR will cover anywhere from 9kHz to 300Ghz in frequency range. Though there are a lot of advantages of SDR, they have a number of technology limitations on the achievable RF performance. The power requirements of SDR tend to be high. Even medium performance SDR require more power for a particular function than equipment specifically designed for that function with optimum digital architectural partitioning. The RF performance obtained through SDRs varies with frequency. The complexity of digital signal processing depends upon bandwidth and demodulation requirements. Maximum supported capability, and in-turn complexity, is limited by power requirements.

The SDR may be more or less complex than the classical approach it replaces, depending upon the range of frequencies and modulation types it has to handle. For very wide frequency range, the RF hardware may need to have a built in separate circuitry for particular frequency ranges. So, the advent of SDR presents new challenges in design, power consumption, measurements and also standards production.

The internal structure of such an SDR (USRP2) is shown in Fig. 4.2. Currently, not all applications are universally covered under SDR technology. There may be technologies, wherein SDR may not offer advantages as that of conventional technology because of complexity, price and power consumption. But, for research and development in the field of wireless communication, this offers a lucrative solution.

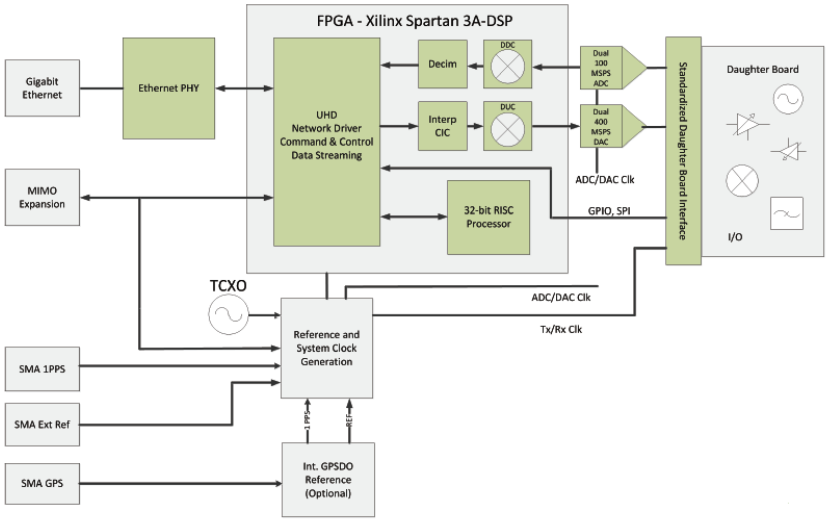

Figure 4.2. Internal Diagram of SDR (USRP2)

4.1. PLATFORM ARCHITECTURE

There are a few open source and proprietary software used for implementation of the communication system. The platform that we use in this section is GNU Radio [30], which is a free and open-source software development toolkit, that provides us with ready-to-use signal processing blocks to implement our algorithms using software radios. It can be either used with low cost external RF hardware or used in a simulation-like environment without hardware. GNU Radio is primarily developed using the GNU/Linux operating system, but, Mac OS and Windows are also supported.

4.1.1. Concept of GNU Radio Flowgraph. GNU Radio[31] is a signal processing package, which is distributed under the terms of the GNU General Public License. In GNU Radio, a radio system is defined as a directed flow graph as in Fig. 4.3, where the graph vertices represent the represent the signal processing blocks, while the edges represent the connections between the blocks. So, in the flowgraph, the data flows from a signal source to one or more sinks. In GNU Radio flowgraphs, the restriction of signal flow cannot form a feedback cycle and hence the implementation of any feedback mechanism must be restricted to a single signal-processing block.

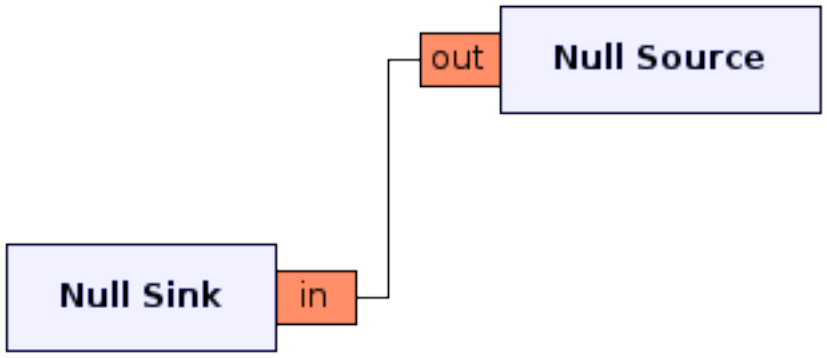

Figure 4.3. Example of a GNU Radio Flowgraph [32]

GNU Radio applications are mainly written in Python as a programming language, while the signal processing blocks are written in C++ using processor floating point extensions. This helps us develop real-time applications. Using high level language like Python[33] as a glue language, helps us in increasing the agility of software development. This is done by offloading most of the heavy processing work to be done by C++ [34] compiled code. Interoperation and data transfer between Python and Swig is managed by the Simple Wrapper Interface Generator (SWIG).

Similar to any programming languages, GNU Radio uses a number of data types to represent the signal at various signal processing block interfaces. A particular block can be identified by the naming convention that is suffixed to its definition. In GNU

Radio, a graphical tool named GNU Radio Companion (GRC) is used, which provides a user interface. This lets us create signal flow graph and activates its source code. The graphical interface allows us to generate a signal flow, and to visualize the signal at every step of the block chain using graphical sinks.

4.2. REQUIREMENTS OF GNU RADIO

The purpose of software-defined radios is to minimize the use of hardware by substituting most of the signal processing to be done by computers. But, they still need to send out and receive signals wirelessly through air as a medium. These conditions constitute as requirements in the field of software defined radios. So, if an application is running continuously as in the case of software-defined radios, there needs to be a processor, which can take care of the signal processing requirements in real time. Each application instant running will have a certain complexity in terms of processing. This translates to a certain requirement. These requirements are highly critical when using a SDR and when implementing it in a platform like GNU Radio.

The requirements of a simple GNU Radio based application are basically a computer, as there is no need for complex operations to be done, and a device to act as the RF interface. However, there are more complicated applications, which need higher computational resources, such as signals and algorithms that act in real time. Also, while operating at wide-band frequencies, there is a need for special circuitry. GNU Radio is meant to work with different hardware that takes care of translation of digital signal into an analog signal centered at a desired frequency. For those applications that operate at higher frequencies, there is a need for an external device. GNU Radio implements the software module for one such device, called the Universal Software Radio Peripheral (USRP). With the help of USRP, it is possible to operate a radio application until a few GHz.

4.3. THE USRP2 PERIPHERAL

USRP, also known as Universal Software Radio Peripheral [35], are basically computer hosted software radios. They are designed and sold by Ettus Research,

LLC. It is responsible for conversion between digital baseband signal generated in the host computer and the analog intermediate frequency. USRPs are also responsible for modulation and demodulation from and to the carrier signal. There are two different versions of USRPs, USRP and USRP2. It depends on the application that any one of them is used. In this work, we use USRP2 that offers various plug-on daughterboards operating on different radio frequency bands.

USRP2 is shown in the Fig. 4.4. It consists of a motherboard with an FPGA that takes care of basic bandwidth calculations and signal processing. There are also two sets of ADCs/DACs to provide a bandwidth of up to 100 mega samples per second (MSps). The FPGA's software can be programmed and stored in an SD card that is inserted into a slot on the front panel. The motherboard is connected to a daughter board that takes care of the transmission and reception of signals at desired frequency. That includes up- and down-conversion of RF signal. Depending on the range of frequencies in which the USRP operates, there are different daughterboards [36]. We can find daughterboards that operate between the frequencies 0 Hz to 5.85GHz. Apart from this, the USRPs also include a power supply and antenna connectors. The daughterboard used for this project is WBX as shown in Fig. 4.5, which operates at frequencies 50MHz to 2.2GHz and has a effective bandwidth of 40MHz.

Figure 4.4. Universal Software Radio Peripheral 2

The FPGA in the motherboard consists of digital up convertors and digital down convertors to convert intermediate frequency signal to baseband signal and vice-versa. It also decimates the signal to achieve data rates compatible with the PC. One of the main differences between USRP and USRP2 is the achievable data rate. Another difference is that the USRP uses USB 2.0 connection to interface with the PC and USRP2 uses a gigabit Ethernet connection. USB 2.0 mainly introduces a bottleneck in applications requiring a higher data rate. Also, USRP2 uses higher precision ADC's/DAC's and also allow simultaneously sending and receiving of data. Also, USRP2 can act as a standalone system because of its high power FPGA.

4.4. MOTIVATION FOR HARDWARE IMPLEMENTATION IN GNU RADIO COMPANION

In our project, we use GNU Radio as a means of testing our multi-variable optimization scheme. We use DS-DBPSK and DS-OFDM schemes to compare and contrast the performance of our algorithm. An extension to GNU Radio is called GNU Radio Companion (GRC) [38]. It provides a graphical interface to create real-time applications. It has a list of available modules, which can be added to an application. These modules have options that can be configured according to application specific notes. This saves us a lot of time in implementing stochastic signal processing algorithms in real-time. It has both advantages and disadvantages.

The main advantage is that the parameters can be easily configured, hence providing a fast-forward method to testing a module. But, the disadvantage is that, there is absolutely no way to customize a module out of the configuration of the parameters. Hence, GRC can be used as a means to start testing different applications and a platform for building complex applications. We study the performance of the system that we want to implement. It will be compared and contrasted for the two specified modulation schemes with the Simulink model in the previous section. This gives us a measure of the maximum achievable performance limits as compared to the theoretical limit.

There are certain real-time requirements that need to be fulfilled to obtain correct behavior of the system. For example, updating the channel estimation rate in

the presence of Doppler effect. These conditions cannot be exactly reproduced. The resources that SDR require are application-specific. The computational resources used for our project implementation include, a Intel I5-core personal laptop running Ubuntu on virtual box software, which ran GNU Radio software and also interfaced with three USRP2 [39].

4.5. GNU RADIO'S DBPSK MODULATOR MODULE

The Differential binary phase shift keying (DBPSK) modulator preforms raised cosine filtering and phase shift modulation of the input we give it. Usually the input to the DBPSK modulator is a random source. If the input data is a byte stream, then the output coming out from the DBPSK modulator is a complex modulated signal at baseband level. The DBPSK module can be subdivided into five different blocks. These blocks are basically written in C++ and interconnected using Python. The signal flow between the sub-blocks is shown as in Fig. 4.6.

The random source generates a binary sequence from a list of byte type numbers, which have no pre-defined timing. The timing here is set at the demodulator block. This is done using parameters including: sample rate and samples/symbol. Root raised cosine filter is used for demodulation purposes and is used for interpolation. The different sub-blocks are explained in detail in the remaining sections.

4.5.1. Packed to Unpacked Block. The random input source gives out packed bytes. It is necessary to convert these into 1-bit per chunk stream for symbol mapping. This block exists in the file *packed_to_unpacked_bb.cc*. This block converts the data depending upon the number of bits per chunk and the endianness. The packed to unpacked format of a 3-bit data is an shown in the Fig. 4.7.

4.5.2. Mapper. The packed to unpacked block provides us with the required bits per chunk. Then, the mapping process is carried out on the chunks. Here, a binary sequence is mapped to symbols. Here we deal with BPSK, it maps 0 and 1 to 0 and 1 respectively. This mapping exists in the file *map_bb.cc*. This mapping is different in the case of QPSK with the gray code chosen, that is, 11 and 10 correspond to 2 and 3 etc.

4.5.3. Differential Encoder. Here a stream of chunks is differentially as,

$$Y_{n+1} = Y_n \bigoplus b_{n+1} \tag{45}$$

An example of such a differential encoder is shown in Fig. 4.8. Here, the binary chunks adds a change in phase with respect to the previous phase of the carrier signal. This is present in the file *diff_encoder_bb.cc*.

4.5.4. Chunks to Symbols. A stream of unpacked bytes is mapped into a stream of float or complex constellation points by this block. This block combined with the previous packed to unpacked block, deals with the general case of mapping from a stream of bytes into arbitrary float or complex symbols. This is present in the *chunks_to_symbols_bc.cc* file.

4.5.5. Root Raised Cosine Filter. The root raised cosine filter is used over a modulated signal. This block is used to reduce the Inter-symbol-interference (ISI), which causes overlapping into the adjacent time slots. This occurs due to time spreading in a real-time system. At the output of this block, we have the complex filtered samples of a baseband carrier ready to be transmitted. This is present in the *root_raised_cosine.cc* file.

4.6. GNU RADIO'S DBPSK DEMODULATOR MODULE

This block performs root raise cosine filtering and a differential coherent detection. This is also known as phase shift demodulation. The input to this block consists of complex sampled signal at baseband and the output is a big-endian stream of packed bits (1-bit per byte). This block is internally sub-divided into 7 sub-blocks, they are shown in the Fig. 4.9. The main parameters needed for this block are the number of samples per symbol, the excess in bandwidth, the Costas alpha parameter, the mu factor and its gain. The different sub-blocks are explained in detail in the remaining sections.

4.6.1. Automatic Gain Control. Automatic gain control (AGC) block, the signal is first multiplied by a constant to scale it from full range to range ±. The automatic gain control process involves a calculation of maximum power level among

the samples. Then, a simple scaling is performed. The AGC works by measuring a smoothed power history, then adjusting the output gain to achieve constant power.

This exists in the file *gr_agc2_cc.cc*Basically, the C++ Algorithm states that the AGC takes a complex sample as input and multiplies it with a gain parameter. Then, the output sample power is calculated. A control loop is made to check the signal power against the reference, until it reaches the desired value.

4.6.2. Frequency Locked Loop. Frequency Locked Loop (FLL) is used for carrier recovery using band-edge filter. The theory of carrier recovery is implemented in C++ algorithm and is present in the file *gr_fll_band_edge_cc.cc*. This is shown in the Fig. 4.10. The Maximum likelihood frequency estimator is used to compensate for frequency offset in the received signal. A filter that is the derivative of the matched filter in frequency domain is constructed. The derivative is zeero everywhere except in the transition band of the filter. Then the product of the input spectrum and the band edge filter is performed and we monitor the two spectral segments in the output, which provides the error signal. This error signal is used to adjust the spectrum of the modulated data.

4.6.3. Poly Phase Filter Bank. The Poly phase filter bank is used mainly for timing recovery. This is present in the file *gr_pfb_clock_sync_ccf.cc*. The samples of the received signal are not exactly aligned with the analog input waveform. To solve this timing problem, we use two filter banks. One filter bank contains the signal's pulse shaping matched filter. Each branch of the filter bank contains a different phase of the filter. Second filter bank contains the derivative of the first filter. Visualizing this in time domain, the output of the first filter bank is shaped like a auto-correlation function. The output signal to be sampled is to be aligned exactly at the peak of the autocorrelation function.

The derivative of the autocorrelation contains a zero at the maximum point of it. The region around the zero point is relatively linear. This is used to generate the error signal. This error signal is used to tune the selection of the filters in the filter bank based on the output of the derivative filter bank as in Fig. 4.11. But, using the derivative filter bank alone is insufficient in correcting the timing.

The solution for the ambiguity with both a positive and negative slope is to get the product of the derivative matched filter output and the matched filter output

as shown in Fig. 4.12. The number of filters in the filter bank is chosen to be 32 filters with 22 taps for each one. Root raised cosine filter taps is generated using the function root_raised_cosine() and the implementation of the function is found in the file *gr_firdes.cc*.

4.6.4. Costas Loop. The main aim of the Costas loop is phase recovery. It helps in correction of the phase error. The implementation of this block is in the file *gr_costas_loop_cc.cc*. The complex received signal is corrected for frequency offset and timing recovery is performed. In BPSK, the real part is the BPSK baseband signal and the imaginary part is the error signal. A positive error needs a phase correction in clockwise direction and the negative error needs a phase correction in the counter clockwise direction. This is shown in the Fig. 4.13.

4.6.5. Differential Decoder. Differential decoding is based on phase change of symbols. Differential decoding is implemented by multiplying each sample of the input by the conjugate of the previous sample. This is done so as to get the phase difference, which encodes the actual data. The implementation of this is in the file *diff_phasor_cc.cc*. The input to this block is the complex received signal after performing the phase, frequency and timing recovery on it. The output is the phase difference.

4.6.6. Constellation Decoder. This is responsible for the decision making process involved in finding the closest constellation point. The decision is taken based on the minimum distance calculated. The implementation of this block is in the file *gr_constellation_decoder_cb.cc*. The input to this block is complex data as phase difference. The decision-making gives out a byte, which represents the corresponding constellation point. The BPSK output would be -1 or 1.

4.6.7. Symbol Mapper. This takes the constellation points from the constellation decoder block and performs symbol mapping. The implementation of this mapping exists in the file *map_bb.cc*. As the input is a group of bytes representing the constellation points, the output is a group of bytes, each byte representing one of the transmitted bits. Any number of phase changes may be used to represent the transmitted bits. With more number of phases, the binary data allows equality.

4.7. GNU RADIO'S OFDM MODULATOR MODULE

The Orthogonal Frequency Division Multiplexing (OFDM) modulation makes use of the orthogonality between its subcarriers to maximize the spectral efficiency. So, the subcarrier frequencies must be orthogonal between each other, that is, all the frequencies must be multiples of the inverses of the symbol duration. The orthogonality helps in reduction of the crosstalk interference between sub-carriers and increases spectrum utilization.

The OFDM modulator as in the Fig. 4.14, modulates an OFDM stream based on the options FFT length, occupied tones, and cyclic prefix length and creates OFDM symbols using a specific modulation scheme (BPSK in our case). The OFDM modulator block mainly consists of 4 blocks. They are OFDM Mapper, OFDM Insert preamble, FFT (Reverse) and OFDM cyclic prefix. The OFDM modulator has a complex modulated signal at baseband at output.

The modulator also includes a send function that takes a payload parameter of size 512 bytes. Then the Send_PKT function sends these data to the first module of the modulator, the ofdm_mapper. The payload is then converted into a data packet using make _packet from the python module ofdm_packet_utils. This calculates and adds a CRC to the header addition.

The OFDM packet contains three fields namely, header, payload and CRC32. This is shown in Fig. 4.15. The header is made of two equal length fields, which is used to identify the packet at the receiver. Each of the two fields is made up of an offset and payload with CRC fields. Moreover, the payload with the CRC fields is scrambled with a pseudo random sequence.

4.7.1. OFDM Mapper. This is a module defined in the file *digital_ofdm_map per_bcv.cc*. The main aim of this block is to map the incoming data into OFDM symbols. It has two outputs, the first is a stream of mapped OFDM symbols and the other is an array of characters (flags) indicating the beginning of the first OFDM symbol to ease the process of adding the Preamble.

4.7.2. OFDM Insert Preamble. This is a module defined in the file *gr_ofdm_insert_preambles.cc*. The main aim of this block is to insert pre-modulated symbols (Preamble) before each payload to help with the synchronization process. This helps especially at the start of the frame. There are four states to insert preamble before the data packet. They are shown as in Fig. 4.16.

4.7.3. Reverse FFT. This is a module defined in the file *gr_fft_vcc.cc*. The main aim of this block is to compute iFFT of the produced symbols with preamble. This block is mainly used to ensure orthogonality of the signals. The basic function of an IFFT is to sum up the individual orthogonal sub-carriers and give as output. Thus this block provides us an easy way to modulate data onto N-orthogonal subcarriers. N output samples from the IFFT block make an OFDM symbol.

4.7.4. OFDM Cyclic Prefix. This is a module defined in the file *gr_ofdm_cyclic_prefixer.cc*. The main aim of this block is to add cyclic prefix to the OFDM symbols. After the Fourier transform from the previous step, the redundant cyclic prefix must be added to the OFDM signal. It takes the OFDM symbol and copies the last symbols at the beginning of the symbol. This creates a symbol with a size, that is the sum of the size of the entered symbol and the size of the cyclic prefix.

4.8. GNU RADIO'S OFDM DEMODULATOR MODULE

The OFDM demodulator as shown in Fig. 4.17, has two basic modules. The first one is the ofdm_receiver, which takes care of the synchronization and the equalization of the CRC signal. The second module is the ofdm_frame_sink, which is a state machine that takes care of the demapping of symbols into bits. This module also checks the validity of the synchronized frames and sends them to the queue of received packets.

The block demodulates a received OFDM stream based on the options like FFT length, occupied tones and CP length. In short, this block performs synchronization, FFT, demodulation of the incoming OFDM symbols and passes the packets to a higher layer.

4.8.1. OFDM Receiver. This module is defined in the file *ofdm_receiver.py*. It basically has four main sub-modules written in C++. They are Channel filter,

OFDM Symbol Synchronization, FFT, and Frame Acquisition. The block diagram of the OFDM receiver is as shown in the Fig. 4.17.

The input to this block is complex modulated signal at baseband. It performs receiver synchronization on the OFDM symbols. It also performs channel filtering as well as symbol, frequency and phase synchronization. Three synchronization routines are available: they are preamble correlator, modified correlator with autocorrelation and cyclic prefix correlator. The first module of the receiver is a channel filter located in the file *gr_fft_filter_ccc.cc*. It is a simple low pass filter for the input coming from the antenna.

The second block is the OFDM symbol synchronization defined in the file *ofdm_sync_pn.py*. This block is responsible for timing synchronization and frequency error correction based on Schmidl and Cox' theory [40]. Here, the synchronization is based on PN sequence correlation. To achieve timing synchronization, good correlation property is required and hence PN sequence is used for preambles.

The third block is the frequency modulator defined in the file *gr_frequency_mo dulator_fc.cc*. The input to this block is the frequency error correction value obtained from previous block. This block is used to generate the frequency offset corresponding to each sample. The fourth block is the OFDM sampler defined in the file *digital_ofdm_sampler.cc*. This block is used to sample the signal coming from the channel using the frequency offset.

The fifth block is the FFT block defined in the file *gr_fft_vcc.cc*. It takes the complex value vector from the sampler and computes the FFT. The last block in the module is the frame acquisition defined in the file *digital_ofdm_frame_acquisition.cc*. This takes the output of the FFT of the received OFDM symbol and finds the start of a frame based on two known symbols. It uses one of the known symbols to estimate the channel over all of the sub-carriers and then performs 1-tap equalization on all the sub-carriers. This helps in correction of the phase and amplitude distortion caused by the channel.

4.8.2. OFDM Frame Sink. The OFDM frame sink is defined in the file *digital_ofdm_frame_sink.cc*. It is a state machine that demaps the symbols into bits and finds the validity of the synchronized frames. The state machine has three states; one is sync search, in which the algorithm looks for the flag indicating the beginning

of the packet. This is shown in Fig. 4.18.

Once the sync is found, it goes into the have sync state. Here, it will let the preamble through. Later, the algorithm demaps the symbols and checks the bytes corresponding to the header of the data. Throughout the duration of the have sync and header states, the algorithm constantly looks for beginning of the frame flags. If an error is found, the state machine is reset to its sync search state.

4.9. DESIGN AND IMPLEMENTATION

There are a number of ways to implement our multi-variable optimization scheme. We use USRP2 (Universal Software Radio Peripheral) as our hardware device and choose a co-operative relaying scheme [41] on our node-setup. The channel that the implementation will use is the air, the signal is transmitted and received using USRP2 antennae. The room of the laboratory where the USRPs are placed will provide multipath echoes. Since, the USRPs will be static, there will no be Doppler effect. The most important case of interferences will be caused in the channel, which is shared with the laboratory's wireless Local area Network (WLAN). The Fig. 4.19 shows the set-up of our project with USRP2s in a triangular pattern.

4.9.1. Setup Using DBPSK Modulation Scheme. We use decode and forward cooperative relaying scheme for the implementing our algorithm. Decode and forward cooperative scheme subsequently reduces the errors in the relay-scheme. We implement our scheme using both DBPSK and also OFDM. Our communication stage is carried out in two time-slots. The first time slot communication is shown in Fig. 4.20.

DBPSK modulation scheme, data is represented by changes in phase of subsequent symbols. It uses gray coding to modulate the encoded bits. In the first time slot, we use a random source to generate packed 1 bit per byte stream and it is spread using PN codes generated by *Galois Linear Feedback shift register* block. Different PN codes are codes are generated by changing the "seed" parameter in the block. It is also the initial state of the feedback shift register. This way, we generate two different sequences destined for relay node and the destination node.The spreading is done by xor'ing the input random bits with PN code.

These spread sequences are added linearly and an equivalent input stream is produced. This is equivalent to Direct Spreading of the sequence. This is then fed into the DBPSK modulator and transmitted from source node (node 1) at IP address 192.168.10.2 to both the relay node (node 2) at IP address 192.168.20.2 and destination node (node 3) at IP address 192.168.30.2. The transmission is done through the USRP Sink block. Reception is through USRP Source Block. All the USRP2s operate at 2.35GHz.

At the receiver, we use the *Automatic Gain Control* (AGC) Block at both the receivers, to compensate for the gain value. The next GRC block implemented is *Polyphase filter bank* block [42].It acts as a timing synchronizer using polyphase filterbanks. This block performs timing synchronization for PSK signals by minimizing the derivative of the filtered signal, which in turn maximizes the SNR and minimizes ISI.

This approach works by setting up two filterbanks; one filterbank contains the signal's pulse shaping matched filter (such as a root raised cosine filter), where each branch of the filterbank contains a different phase of the filter. The second filterbank contains the derivatives of the filters in the first filterbank. Thinking of this in the time domain, the first filterbank contains filters that have a sinc shape to them. We want to align the output signal to be sampled at exactly the peak of the sinc shape. The derivative of the sinc contains a zero at the maximum point of the sinc (sinc(0) = 1, sinc(0)' = 0). Furthermore, the region around the zero point is relatively linear. We make use of this fact to generate the error signal. Hence this block fixes the clock mismatches between the transmitter and receiver.

The next block is the Constant Modulus Adaptive (CMA) Equalizer block [43], that tries to invert multipath so that the combination is a flat frequency response. The error value and tap update equations can be found in [43]. For DBPSK, the constellation diagram before and after timing and multipath corrections is shown in Fig. 4.21. The figure on the left shows the constellation diagram before timing recovery and equalization. The figure on the right shows the corrected one. This shows that even with noise from the laboratory surroundings, it is possible to recover proper timing.

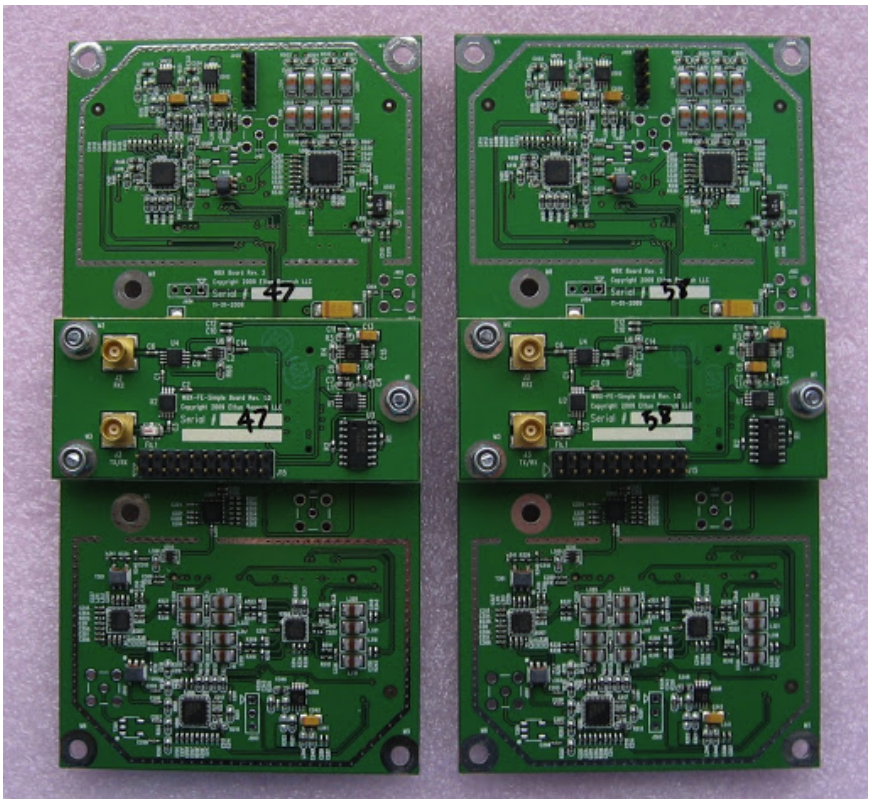

Figure 4.5. WBX Transceiver Daughterboard [37]

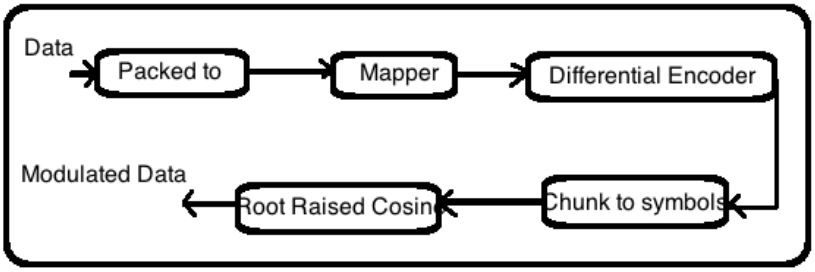

Figure 4.6. DBPSK Modulation Block data flow

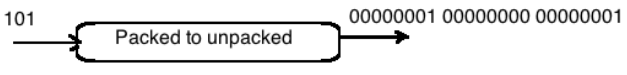

Figure 4.7. Example of packed to unpacked block

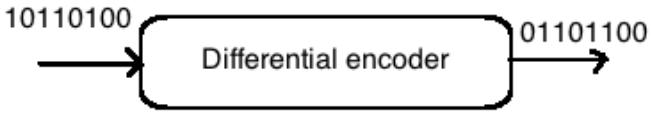

Figure 4.8. Example of differential encoder

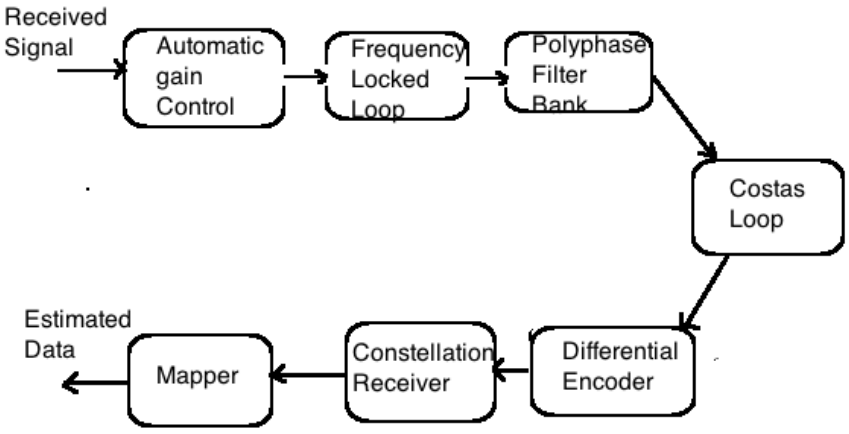

Figure 4.9. DBPSK Demodulation Data Flow

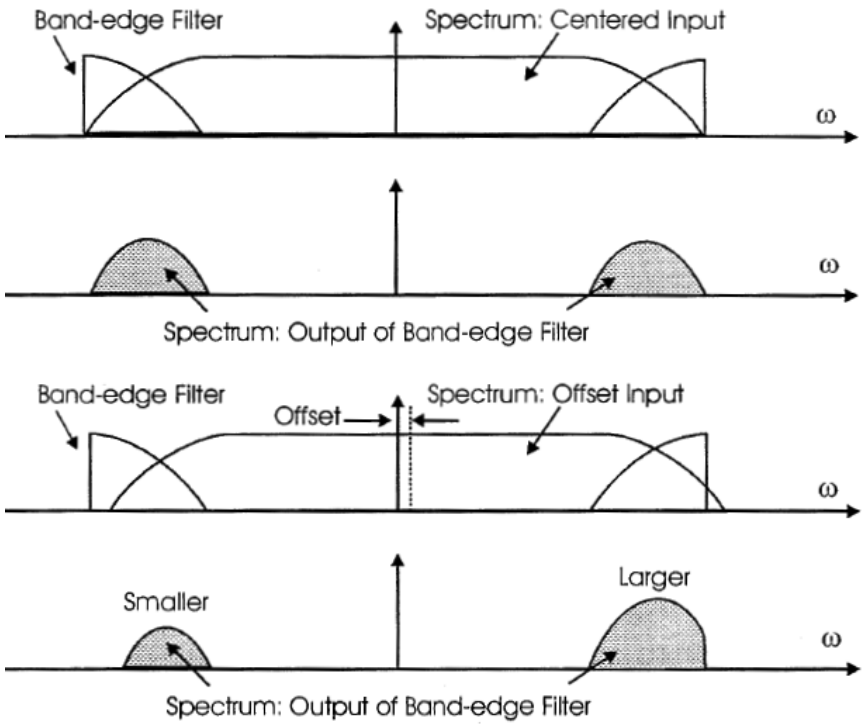

Figure 4.10. Carrier Recovery Process

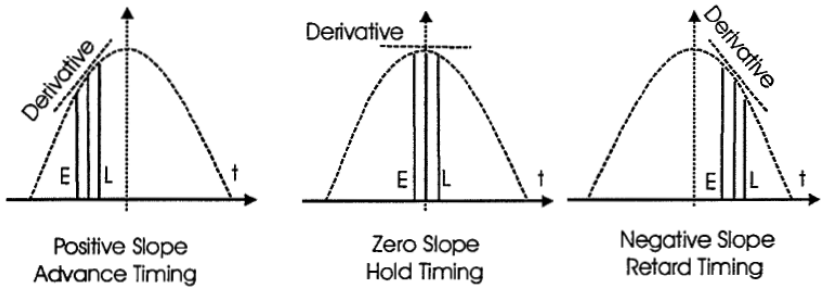

Figure 4.11. Estimating error process

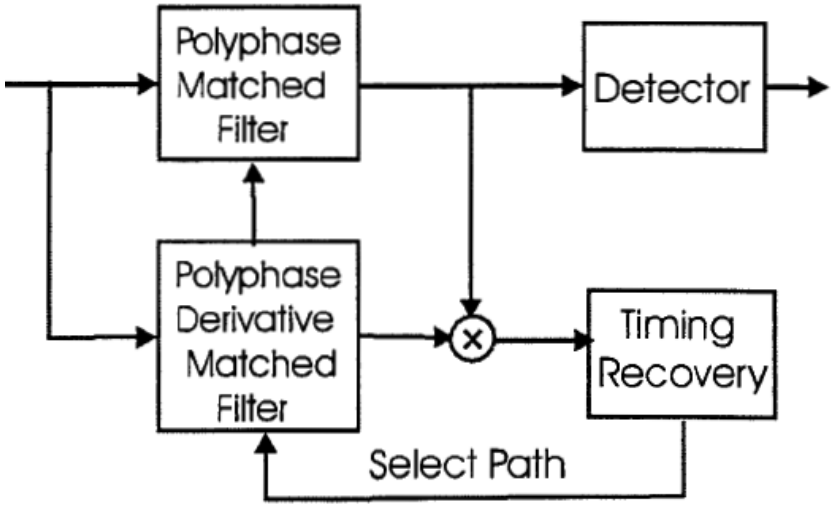

Figure 4.12. Timing Recovery System

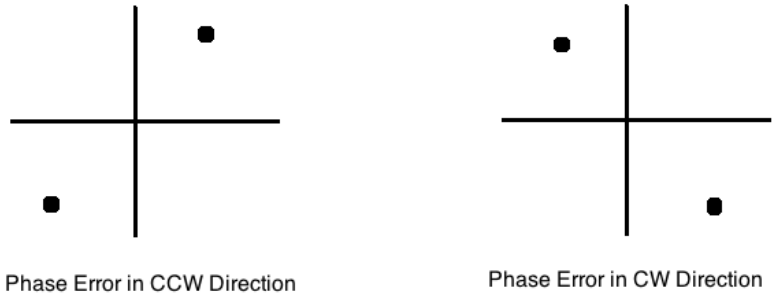

Figure 4.13. Phase Error

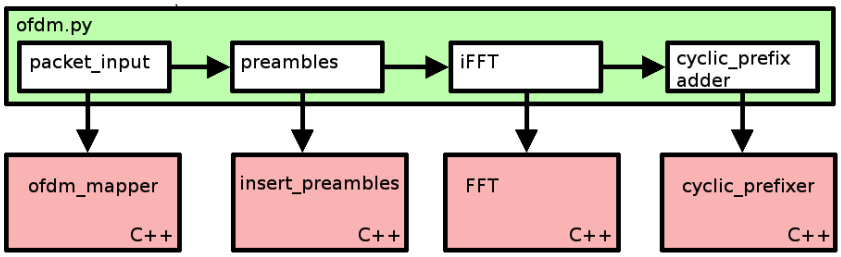

Figure 4.14. OFDM Modulator Hierarchy

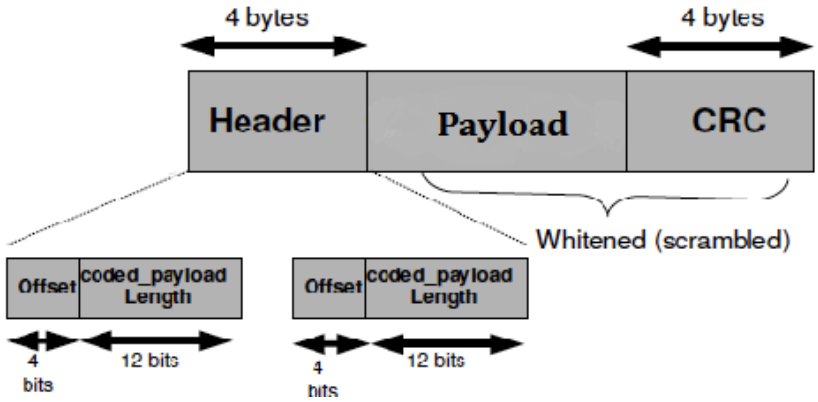

Figure 4.15. OFDM Packet Structure

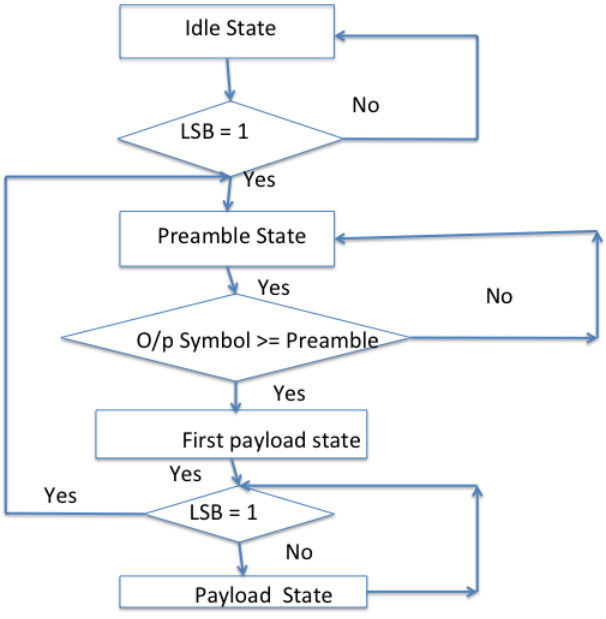

Figure 4.16. OFDM Insert Preamble Flowchart

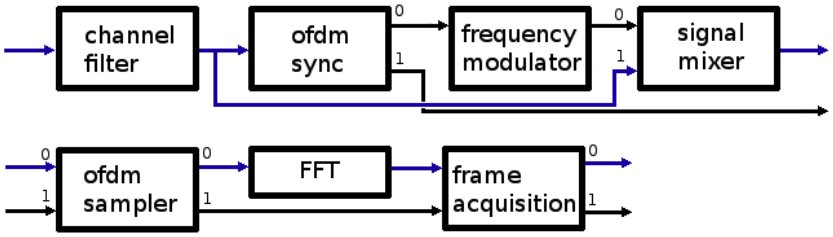

Figure 4.17. OFDM Receiver Block Diagram

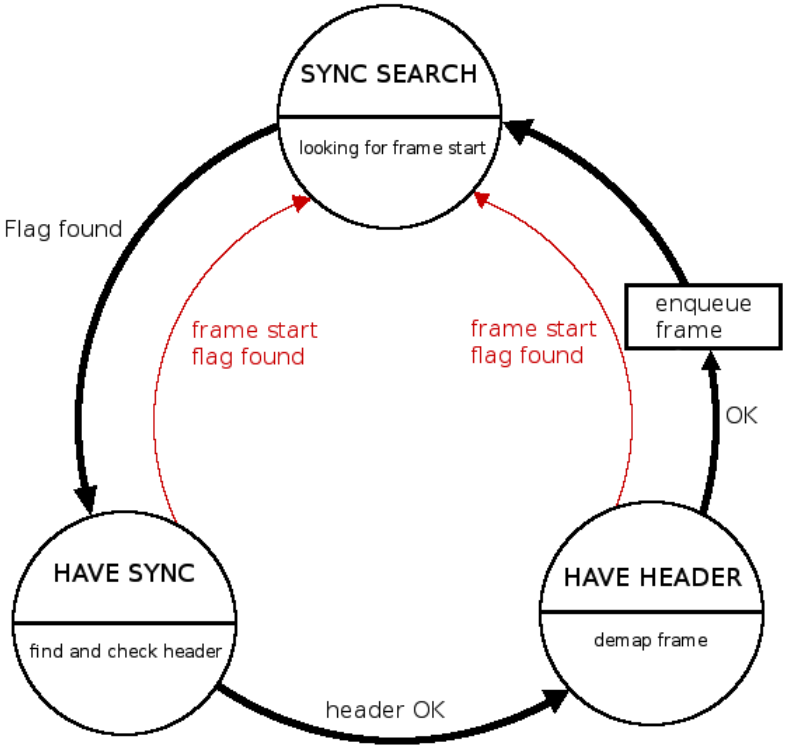

Figure 4.18. OFDM frame Sync State Machine

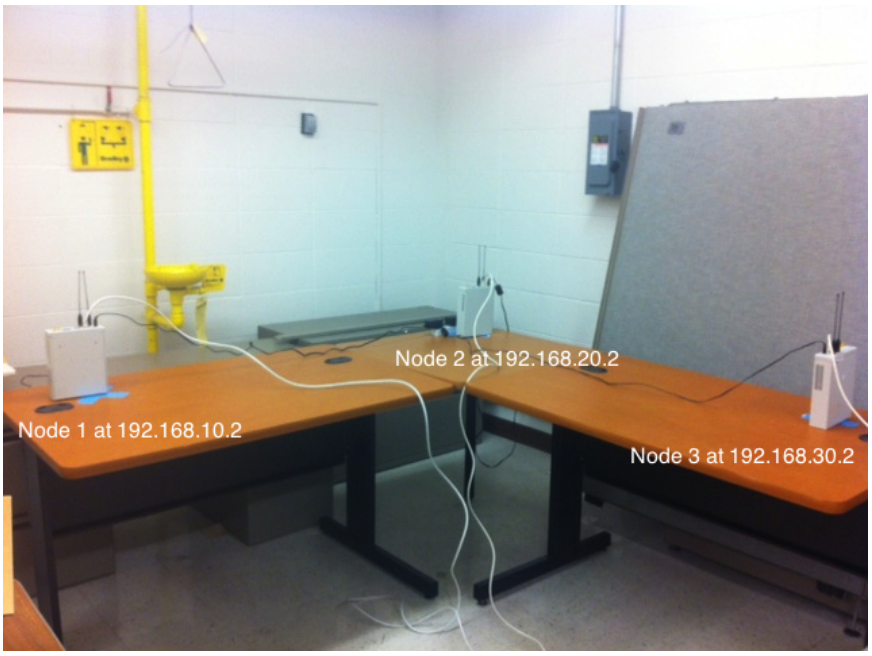

Figure 4.19. Experimental Setup

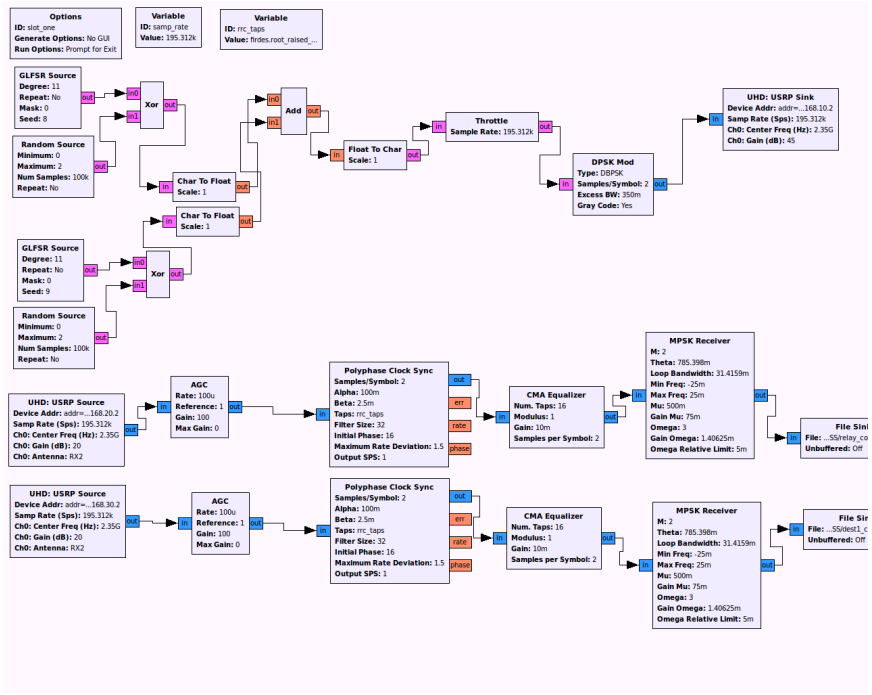

Figure 4.20. Communication in Slot one

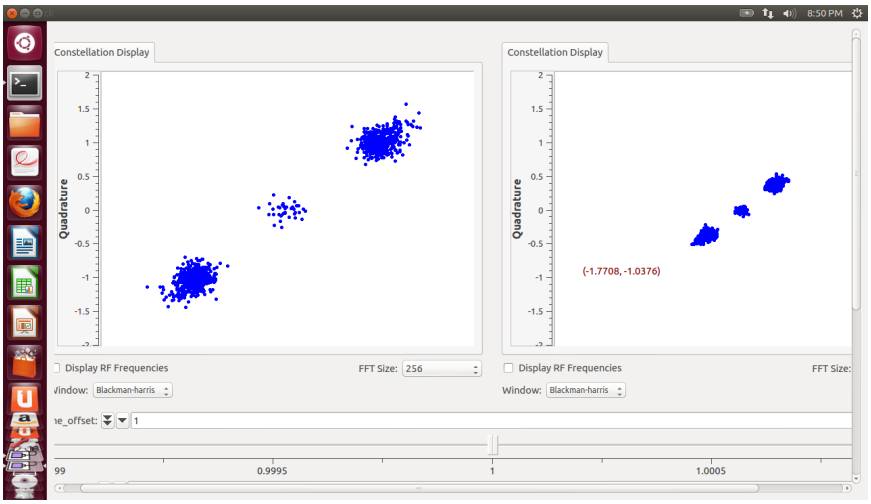

Figure 4.21. Constellation Diagram after timing recovery and equalization

Next is the MPSK receiver block that contains both the demodulation of the data as well as Costas loop, phase, frequency, and symbol synchronization. This block takes care of receiving M-PSK modulated signals through phase, frequency, and symbol synchronization. It performs carrier frequency and phase locking as well as symbol timing recovery.

The phase and frequency synchronization are based on a Costas loop that finds the error of the incoming signal point compared to its nearest constellation point. The frequency and phase of the NCO are updated according to this error. The symbol synchronization is done using a modified Mueller and Muller circuit from the paper [44]. This circuit interpolates the down-converted sample (using the NCO developed by the Costas loop) every mu samples, then it finds the sampling error based on this and the past symbols and the decision made on the samples. The modifications to the M and M used here reduce self-noise.

Hence, the received complex 32 bit data at node 2 is stored as "relay_complex.dat" and at node 3 as "dest1_complex.dat" respectively. This is further processed in the file "phase1_decode.grc" to obtain raw data in bit form. In this phase, the received data is de-spread using the initial used PN sequence for that particular link. We use two different PN codes to differentiate between stream related to node 2 and node 3 respectively. This is shown in Fig. 4.22.

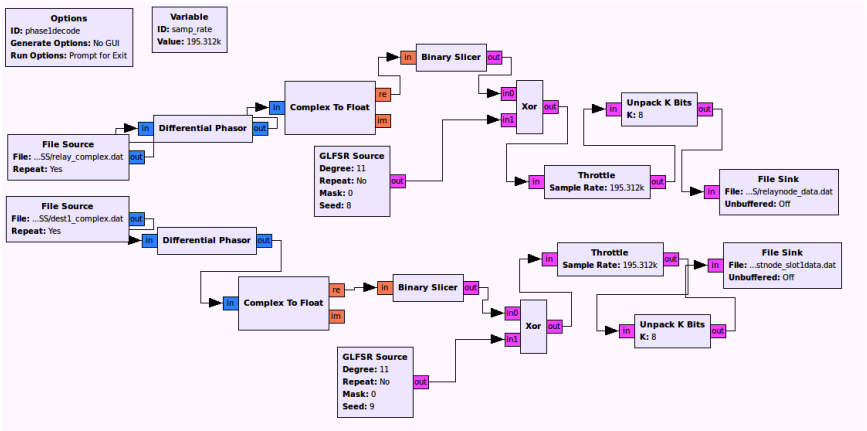

Figure 4.22. Decoding the relay signal and processing

This decoded data is then stored further in .dat files as "relaynode_data.dat", "destnode_slot1data.dat" at nodes 2 and 3 respectively. These files are further used for processing in GRC and Octave for plotting and finding the bit error rate.

The second slot of communication Fig. 4.23 involves re-transmitting the information from node 2 to node 3 by re-encoding and modulating it. At the same-time, node 1 transmits its information to node 3. Thereby, acting as simultaneous transmission and interference source. Finally at the destination node, that is, node 3, we receive the signal and perform the same synchronization steps as in slot 1.

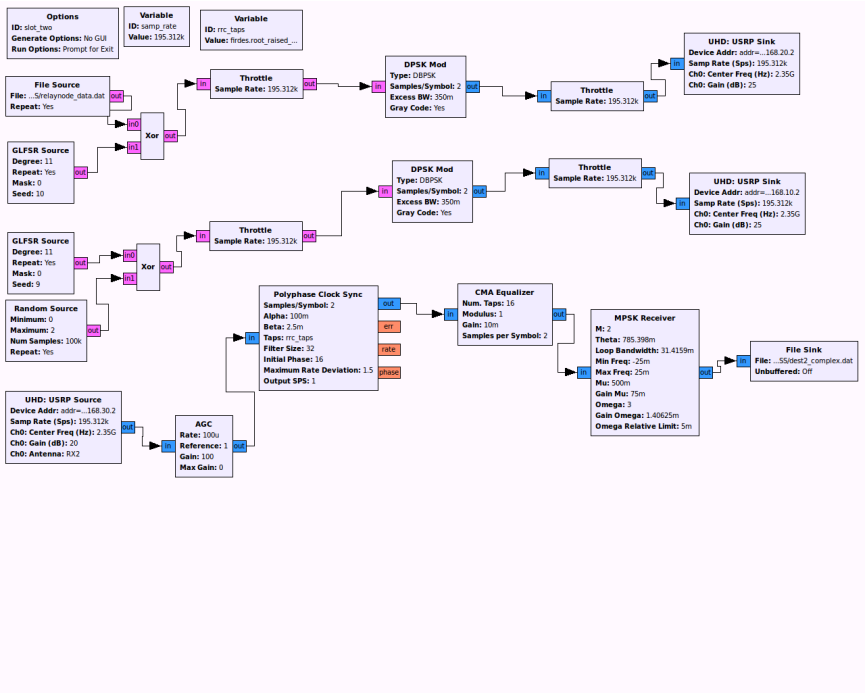

Figure 4.23. Communication in Slot Two

Finally, the signal complex signal stored in "dest2_complex.dat" is decoded in "phase2_Decode.grc" Fig. 4.24 at node 3. This decoded raw file is stored in "destnode_slot2data.dat". Finally, we need a way to add the signals received at node 3 in both the slots. This is done in the file "calculation.grc" as shown in Fig. 4.25. As the signals have already been corrected for timing, phase and frequency errors,

we linearly add the signals in "dest1node_data.dat" and "dest2node_data.dat" and compare it with the initial input data sequence.

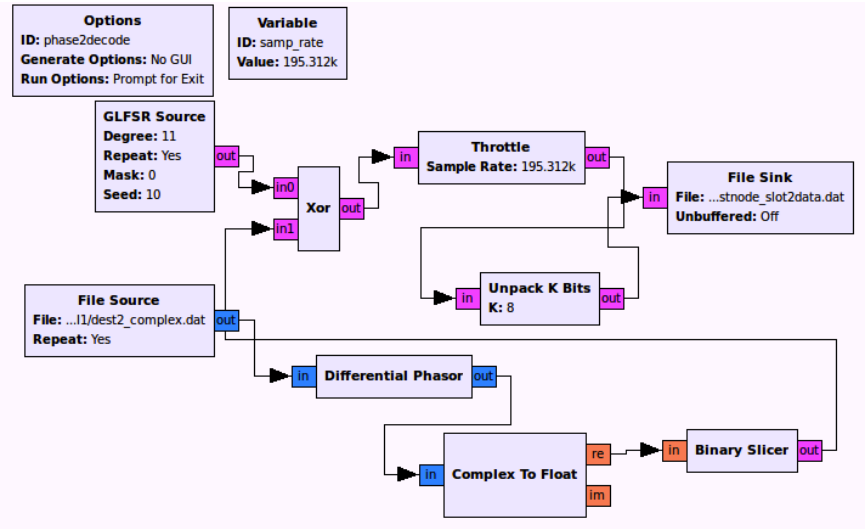

Figure 4.24. Decoding the signal at destination and processing

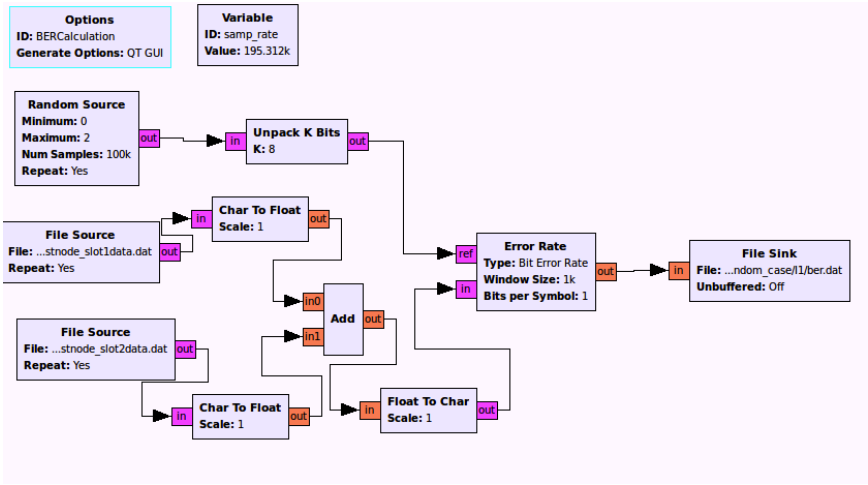

Figure 4.25. Bit Error Rate Calculation

The comparison is done using the Bit Error rate block. The block calculates the error bits with respect to the total number of transmitted bits for that particular window size we key in. The calculated bit error rate is stored in the file "ber.dat" for further processing in Matlab and Octave.

4.9.2. Setup Using OFDM Modulation Scheme. The USRP2 are setup the same as shown in the section 4.19. The communication here happens in two slots as in the previous section. The communication in the first slot is shown as in Fig. 4.26.

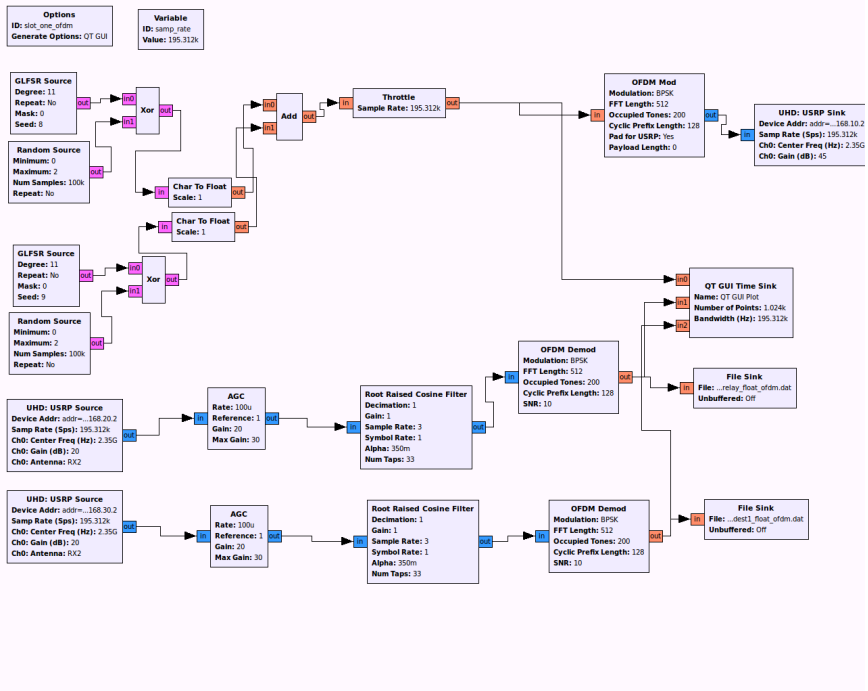

Figure 4.26. Communication in Slot one

At node 1, the random data bits destined for node 2 here get spread like in the previous case and then they get added to another equally spread data destined for node 3. This is then passed on to the OFDM modulator block. As discussed in section 4.14, the OFDM modulator block modulates a stream based on FFT length, occupied tones and cyclic prefix length. This block outputs OFDM symbol based on a specific modulation scheme (BPSK scheme here).

This is then transmitted using USRP sink of node 1 at IP address 192.168.10.2 to USRP source block at node 2 and node 3. At each receiving node, the OFDM demodulator block performs the synchronization before decoding. This eliminates the need for a separate synchronization block.

Hence, the received complex 32 bit data at node 2 is stored as "relay_complex _ofdm.dat" and at node 3 as "dest1_complex_ofdm.dat" respectively. This is further processed in the file "phase1_decode_ofdm.grc" to obtain raw data in bit form. In this phase, the received data is de-spread using the initial used PN sequence for that particular link. We use two different PN codes to differentiate between stream related to node 2 and node 3 respectively. This is shown in Fig. 4.27.

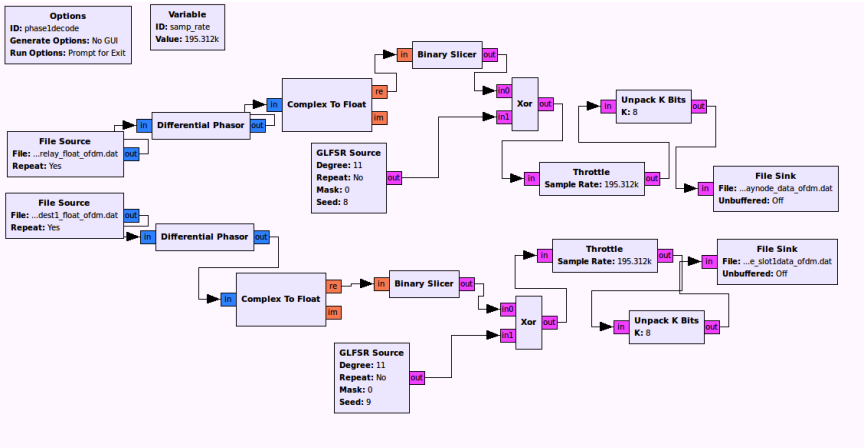

Figure 4.27. Decoding the ofdm relay signal and processing

This decoded data is then stored further in .dat files as "relaynode_data_of dm.dat", "destnode_slot1data_ofdm.dat" at nodes 2 and 3 respectively. These files are further used for processing in GRC and Octave for plotting and finding the bit error rate.

The second slot of communication Fig. 4.28 involves re-transmitting the information from node 2 to node 3 by re-encoding and modulating it. At the same-time,

node 1 transmits its information to node 3. Thereby, acting as simultaneous transmission and interference source. Finally at the destination node, that is, node 3, we receive the signal and perform the same synchronization steps as in slot 1.

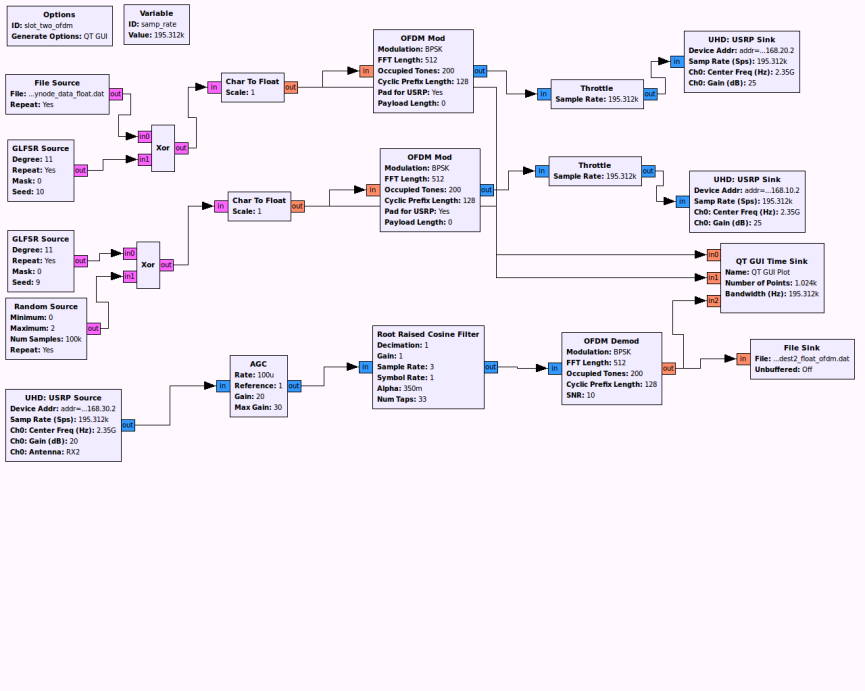

Figure 4.28. Communication in Slot Two - OFDM

Finally, the signal complex signal stored in "dest2_complex_ofdm.dat" is decoded in "phase2_decode_ofdm.grc" 4.29 at node 3 as shown in Fig. 4.29. This decoded raw file is stored in "destnode_slot2data_ofdm.dat". Finally, we need a way to add the signals received at node 3 in both the slots. As the signals have already been corrected for timing, phase and frequency errors, we linearly add the signals in "dest1node_data.dat" and "dest2node_data.dat" and compare it with the initial input data sequence. The comparison is done using the Bit Error Rate (BER) block. The block calculates the error bits with respect to the total number of transmitted bits for that particular window size we key in. The calculated bit error rate is stored in the file "ber.dat" for further processing in Matlab and Octave.

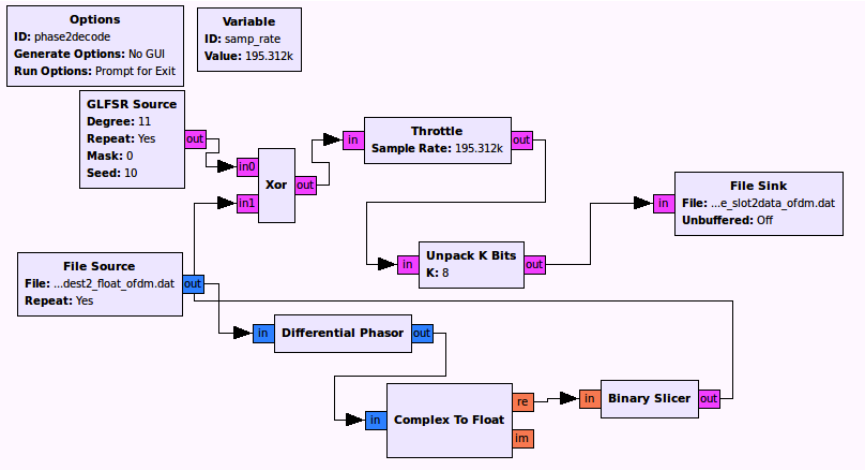

Figure 4.29. Decoding the OFDM signal at destination and processing

5. SIMULATION RESULTS AND ANALYSIS

The simulation is carried out through MATLAB in such a way as to emulate a dynamically changing environment. The three nodes are placed in an environment where the channel characteristics have an uncertainty in them. This occurs basically as multiples of power transmitted.

The channel gain or attenuation formula is as follows:

$$g = f(d, n, X, Si) = d^{-n}\ 10^{0.1.si}\ X^2 \tag{46}$$

Here, d^{-n} is the effect of path loss, $10^{0.1.si}$ represents the effect of shadowing, X is a random variable with Rayleigh distribution and g represents the channel gain varying with uncertainties.

Using, the above formula for dynamic gain generation, we optimize the power requirements for optimal net throughput.For the simulation, we use n=4 representing urban environment and randomly generate the shadowing effect and Rayleigh distribution functions.

The simulation is performed to show the allocation of power and capacity for 3-hop network for which we have defined the terms. The allocation is then carried out for every next three hop network by propagating the effects of the previous 3-hop onto the next 3-hop. The first 3-hop network 1-2-3 is taken as the scenario to be analysed.

Case one describes the optimal power variation P_{12} when P_{13} and P_{32} are equated to a constant P_{max}. Case two describes the optimal power variation P_{13} when P_{12} and P_{32} are equated to a constant P_{max}. Case three describes the optimal power variation P_{12} and P_{13},when P_{32} is equated to a constant P_{max}.

As this simulation and the theoretical analysis is used as a basis for hardware implementation, we place the first three nodes the same way as it is placed in the real-time laboratory situation. The three nodes are placed as in an equilateral triangle with distance between them as 50 inches or 1.27m each. We use MATLAB to generate this particular simulation scenario and find out the optimal power allocation for a particular topology and its variations. Of all the three case scenarios of the first hop,

only case 3 gives out power allocations according to our algorithm. Only case three, where we vary P_{12} and P_{13}, while we keep P_{32} at a constant maximum of 70mw, gives out the optimal power specifications.

The placement of the first hop nodes (node1, node2 and node 3) is shown in the table 5.1. The table shows the placement coordinates in three particular scenarios. The three scenarios include equally spaced nodes, node 1 displaced from its initial position towards the other nodes and away from the other nodes. These three scenarios test the validity of the multi-variable optimization scheme with respect to all corner cases. The Tab.5.2, 5.3, 5.4 and 5.5 give the individual parameter values based on the scenarios described in table .5.1. They give a comparison of the achieved capacity with reference to shortest hop routing and split multipath routing schemes. Here, the values are given for the variables which are set by the optimization algorithm (here Tx-P32, Tx-P34, Tx-64 are all set to maximum by definition of the algorithm).

Table 5.1. Node Iteration Coordinates for first hop

Scenario	(x1 y1)	(x2 y2)	(x3 y3)	Description
Iteration 1	(6 6)	(10 3)	(10 9)	Equally spaced first hop
Iteration 2	(4 7)	(10 3)	(10 9)	Node 1 moving away
Iteration 3	(8 5)	(10 3)	(10 9)	Node 1 moving

Table 5.2. Transmit Power Values Measurement

Scenario	Tx-P12	Tx-P13	Tx-P36
Case 1	13 dbm	6 dbm	(6.2 - 7.6) dbm
Case 2	15 dbm	9 dbm	(6.2 - 7.6) dbm
Case 3	16.5 dbm	9 dbm	(6.2 - 7.6) dbm

The simulation results for the first hop shows these three different corner cases as scenarios and plots them for different iterations of varying multipath scenarios. The simulated results are shown in the figures 5.1, 5.2 and 5.3.

Table 5.3. Signal to Noise Plus Interference Ratio Values Measurement

Scenario	SINR (P12)	SINR (P13)	SINR (P36)
Case 1	-15db	-8db	-17db
Case 2	-16db	-9db	-17.5db
Case 3	-16.5db	-9db	-18db

Table 5.4. Traffic Split Measurement Table

Scenario	Mod-rate	traffic-split 1 Aggregate	traffic Split 2 Aggregate
Case 1	2.2Mbps	7.05Mbps	6.22Mbps
Case 2	2.2Mbps	6.61Mbps	5.76Mbps
Case 3	2.2Mbps	6.005Mbps	5.50Mbps

Table 5.5. Achieved Capacity Comparison Table

Scenario items	Achieved Capacity	Comp Ratio Shortest Path	Comp Ratio Split Multipath
Case 1	6.6 Mbps	15.5 (ratio)	10 (ratio)
Case 2	6.105 Mbps	15 (ratio)	8 (ratio)
Case 3	5.55Mbps	12 (ratio)	7 (ratio)

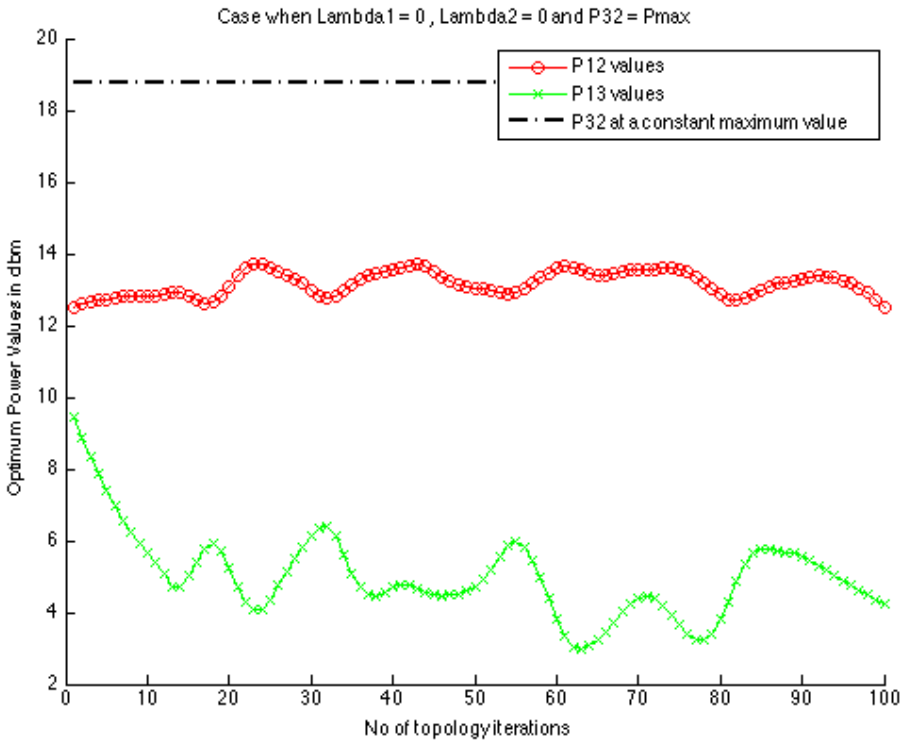

Figure 5.1. Optimal Power Values for Case 3 of first hop (Scenario 1)

The plot in Fig. 5.1 shows that the optimal power allocation for one particular scenario of first hop. It shows that the optimal power ranges between 11 dbm and 18.45 dbm. There are three variations of this particular topology shown and the remaining two are shown in Fig. 5.2 and in Fig. 5.3 in 2D space. The variations describe the movement of node 1 with nodes 2 and 3 static. This helps in deliberately changing the load balancing process made from node 1 to node 2 and node 3. The amount of power required to transmit useful information in a interference constrained environment varies according to the source node (node 1 here) position. The position here matters because the source of information entry is only through node 1.

Once the information has been propagated through the first hop, it has to go through the second hop. This is our topology information flow. As we observed, for the first hop, case 3 is the only optimal case. The optimal case of first hop combined with the optimal case of second hop, gives us our solution. For the second hop, the optimal case would be case 1 (where $\lambda_1 = 0$ and P_{34}, P_{64} are each set to P_{max}. The Fig. 5.4, Fig. 5.5 and Fig. 5.6 show the optimal power allocation for case 1 of second hop combined with case 3 of first hop. The three figures show the appropriate combination of the three scenarios of the first hop with the single case of second hop respectively.

Here, the optimal power values P_{36} for the second hop change between 6.2dbm (4.2mw) and 7.6dbm (60mw). The abrupt changes in the values show the effect of channel gain/attenuation and also the interference between the nodes. We also notice that all the three cases of the second hop reflect identical P_{36} power value range. The abrupt changes in the value are as a result of interference and also movement of node 1, which in turn alters the load distribution. The simulation uses an interference constrained wireless medium and hence tries to emulate a realistic scenario.

The main aim of our algorithm is to implement fair end-to-end throughput or bit error rate in a topology of our choosing. So, we use Simulink package of Matlab as a medium to do this. In here we use a combination of BPSK with spreading with PN code. The combination is found to be robust against channel impairments in a multi-user system. The Fig. 5.7 shows two sub-blocks connected and their respective bit error rate displays. Each sub-block in blue refers to a three-hop network.

The sub-system in simulink is exactly designed as in our topology and the bit-error rate from first hop is propagated to the next. Each subsystem (each 3-hop network) is as shown in Fig. 5.8 and Fig. 5.9 respectively. Random binary data is generated by the Random Integer block. This is then modulated by the BPSK modulator. PN sequence generator generates the spreading code. It then spreads the modulated data. This spread signal is then added to the same kind of data spread by a different PN code generated by another transmitter and this is done by the signal combiner. This combined signal is then sent through the AWGN channel.

On the receiving side, the signal goes through the same despreading process. But each signal goes through the despreading with their respective PN codes. They are then demodulated and then they go through their respective process to calculate the bit-error rate. At the channel, we use a multipath propagation block to send the signal through multiple paths. This allows us to have a realistic scenario and also helps us achieve the spacing effciiency between the nodes. To compensate for the delay we send through the channel and also to properly receive the signals without the timing delay, we use a Rake Combiner as shown in Fig. 5.10.

We implement our algorithm using DS-BPSK instead of higher order modulations is because, the more the number of phases we use, the more bit-error rate we have to deal. To smplify the process of analysis and also to increase our chances of a better throughput, we use the most basic form of PSK (M =2) along with the spreading scheme.

The simulation results show that signal spreading with BPSK is suitable for short distance communication. We observe that the spreading makes the system able to perform better at higher SNR levels. This implies that the higher order modulation cannot handle smaller distance transmission. From the simulation scheme, we observe that by selecting BPSK with multipath errors, timing delays and hopping to different areas, we have to bear with the poorer BER. One has to observe the trade-off in SNR with respect to the BER. For the adopted scheme with multipath environment and simultaneous tranmissions, the saving in SNR we achieve is very high.

From Figs 5.11, 5.12, 5.13, one can easily observe the BER improvement over decreasing symbol rate and improved multipath conditions. The different topology placements indicate that with the SNR levels chosen from our scheme for that par-

ticular distance between the nodes, the BER levels can be improved with improved multipath and timing delay conditions. We also observe that even with the nodes transmitting at the same time and high multipath conditions, the BER rate curve follows the theoretical BPSK model as multiples of BER.

From the simulation results we see that, even though we see a bit error rate in the order of 10^{-3} with multipath requirements, we achieve around 5db, 10db of saving in SNR. After spreading with higher PN sequence, we obtain a lower bit error rate even for bpsk and as the signal is spread across bandwidth of the spreading sequence, we obtain a higher gain.

5.1. OBSERVATIONS AND CONCLUSIONS

The proposed multi-path routing scheme renders an optimal network performance in terms of maximum throughput under constraint of fair allocation of resources to each flow. Moreover the solution is achieved for lowest transmission power required. The theoretical derivation guarantees the optimality of the solution, while the simulation results validate them. Hence, the proposed multi-path routing scheme provides us an improvement over the previous fixed node power split case. It shows that by proper selection of initial power values for optimization, there is tremendous improvement in the achieved net throughput. Future work will include generalization of the routing strategy for arbitrary number of alternative routing paths and traffic flows and also implementation of the scheme in real-time hardware. Moreover, end-to-end optimality will be studied for large networks.

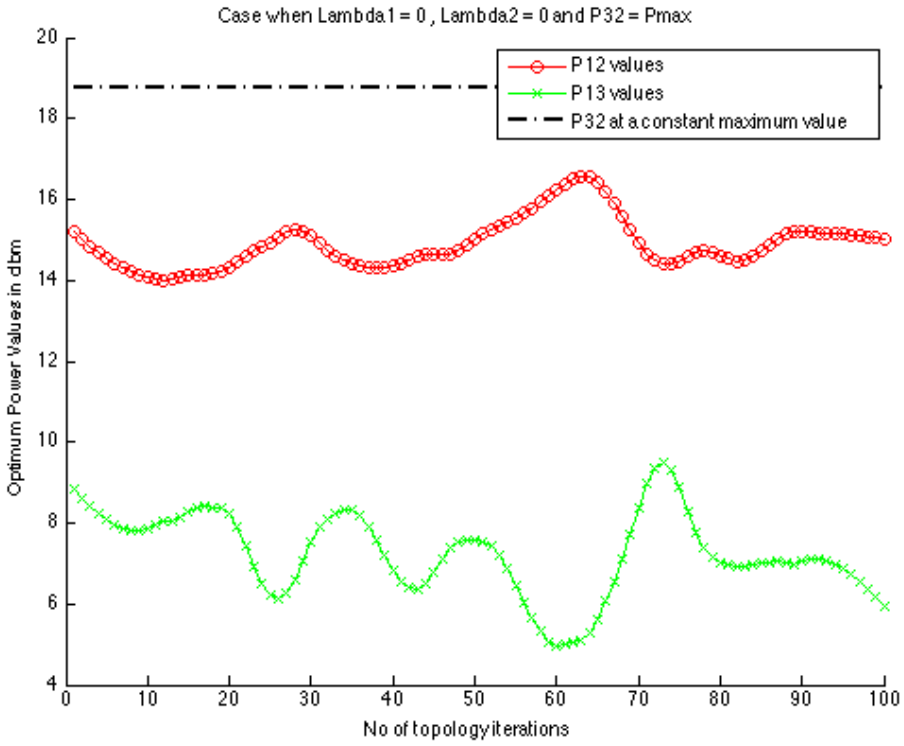

Figure 5.2. Optimal Power Values for Case 3 of first hop (Scenario 2)

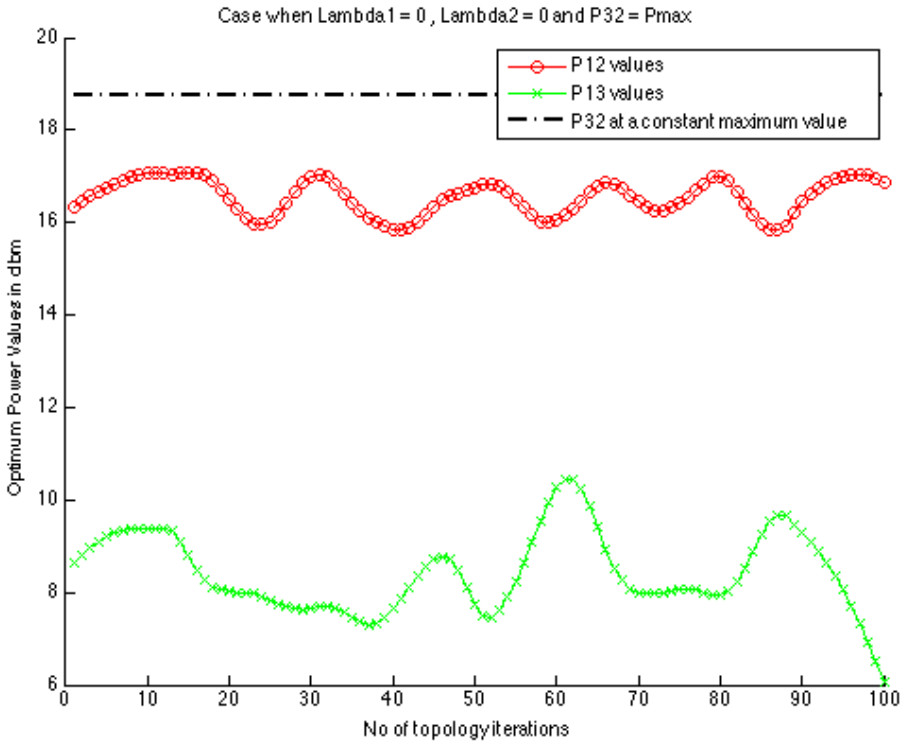

Figure 5.3. Optimal Power Values for Case 3 of first hop (Scenario 3)

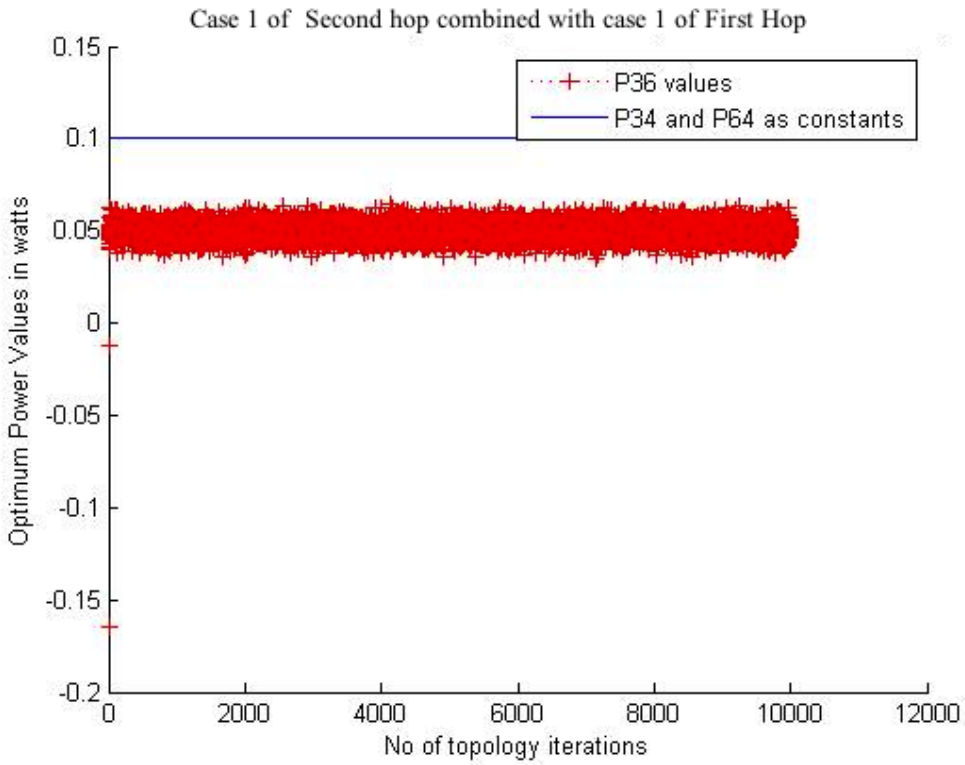

Figure 5.4. Power Values (Scenario 1 of first hop combined with second hop)

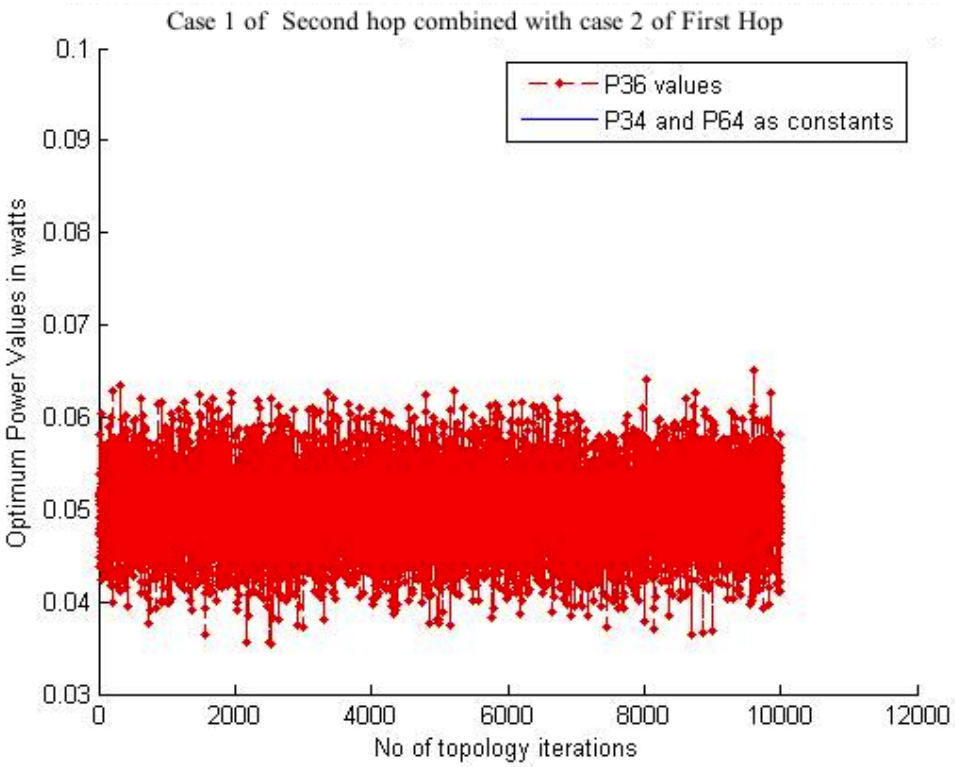

Figure 5.5. Power Values (Scenario 2 of first hop combined with second hop)

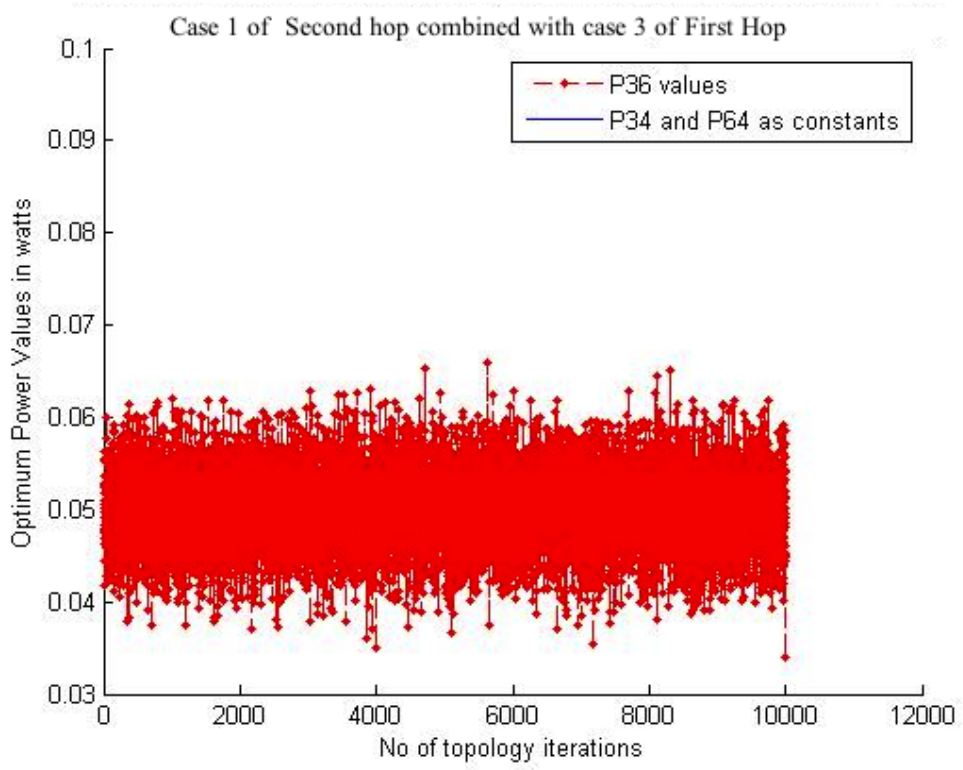

Figure 5.6. Power Values (Scenario 3 of first hop combined with second hop)

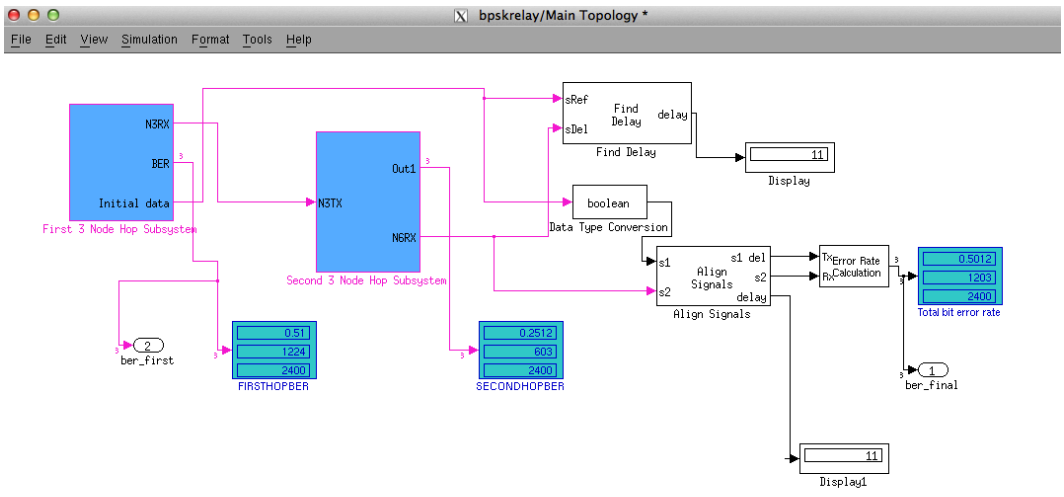

Figure 5.7. Simulink Scheme for DS-BPSK Implementation

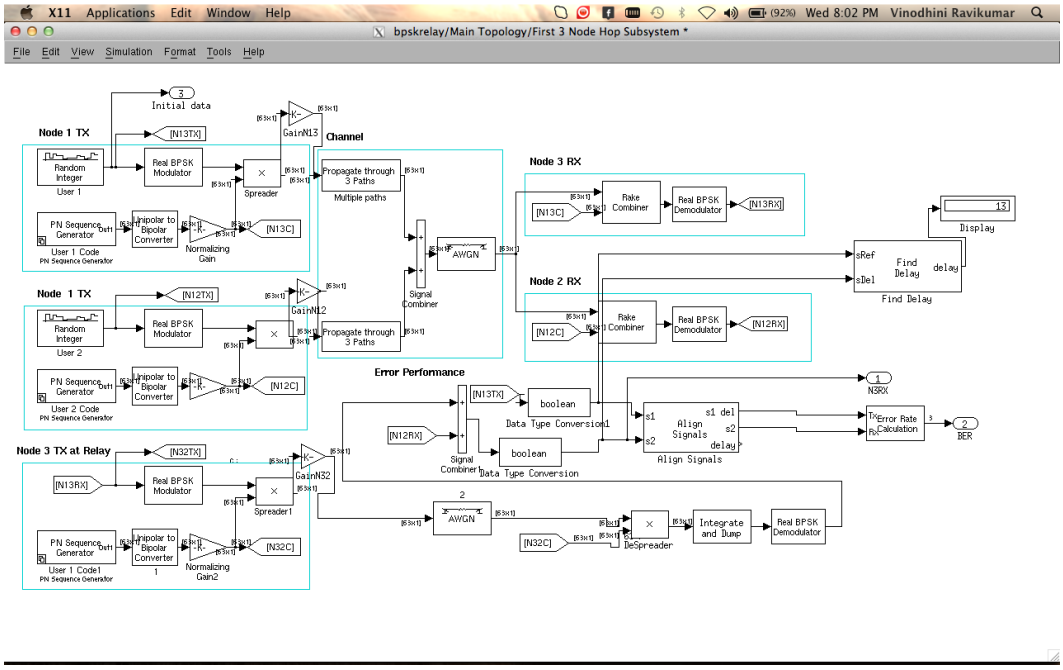

Figure 5.8. First 3-hop network sub-block in the Simulink Scheme

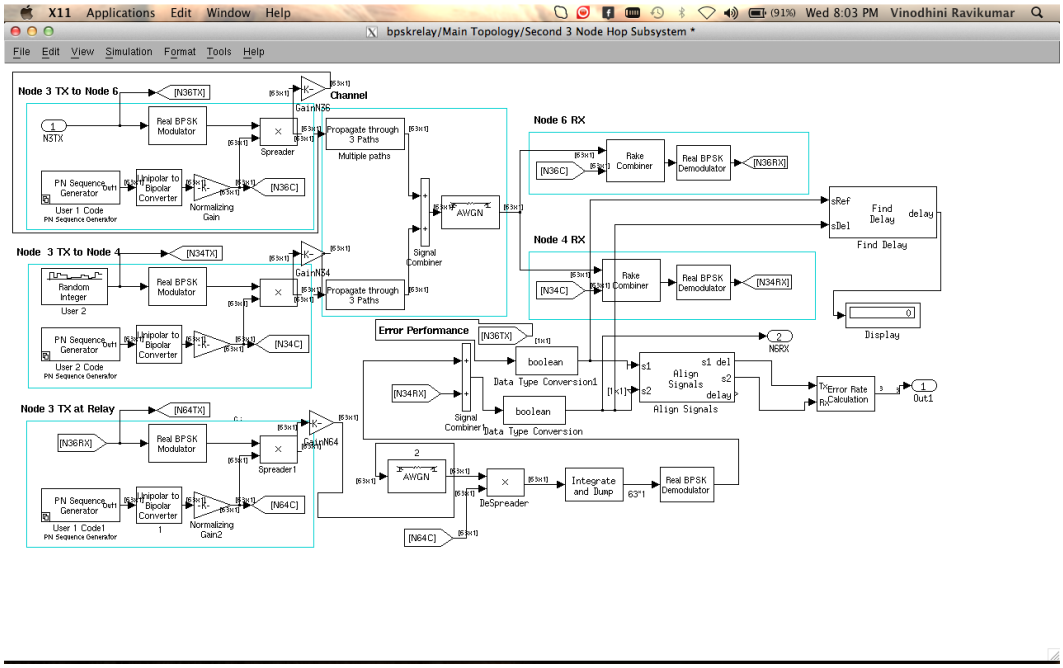

Figure 5.9. Second 3-hop network sub-block in the Simulink Scheme

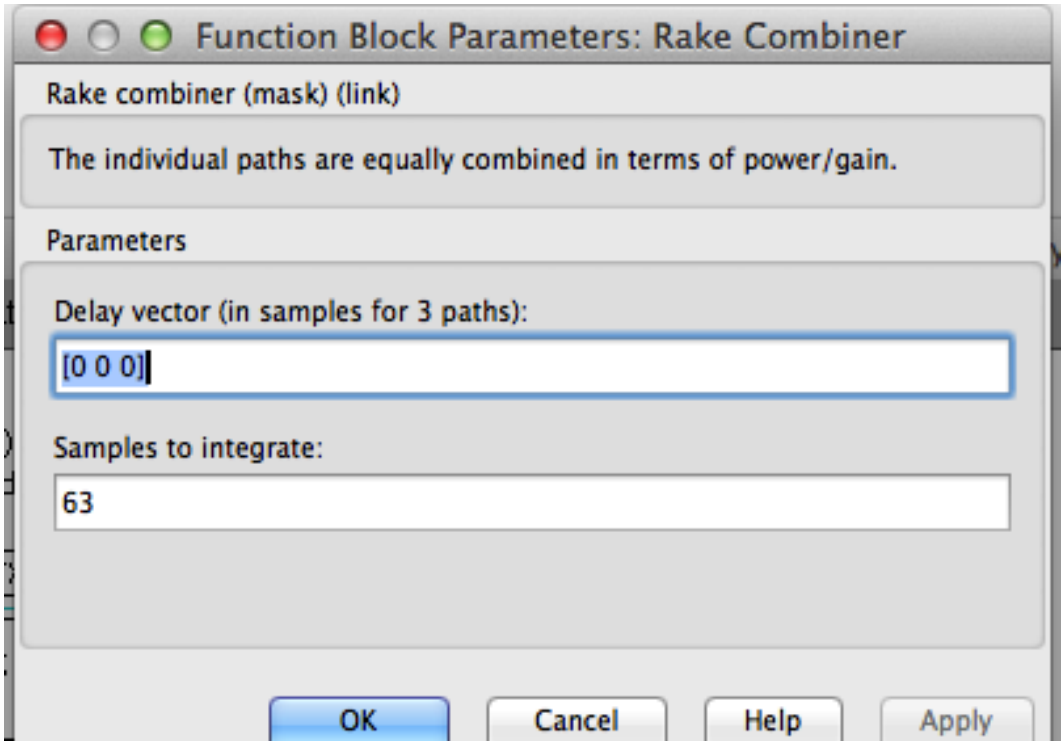

Figure 5.10. Rake Receiver dialog box

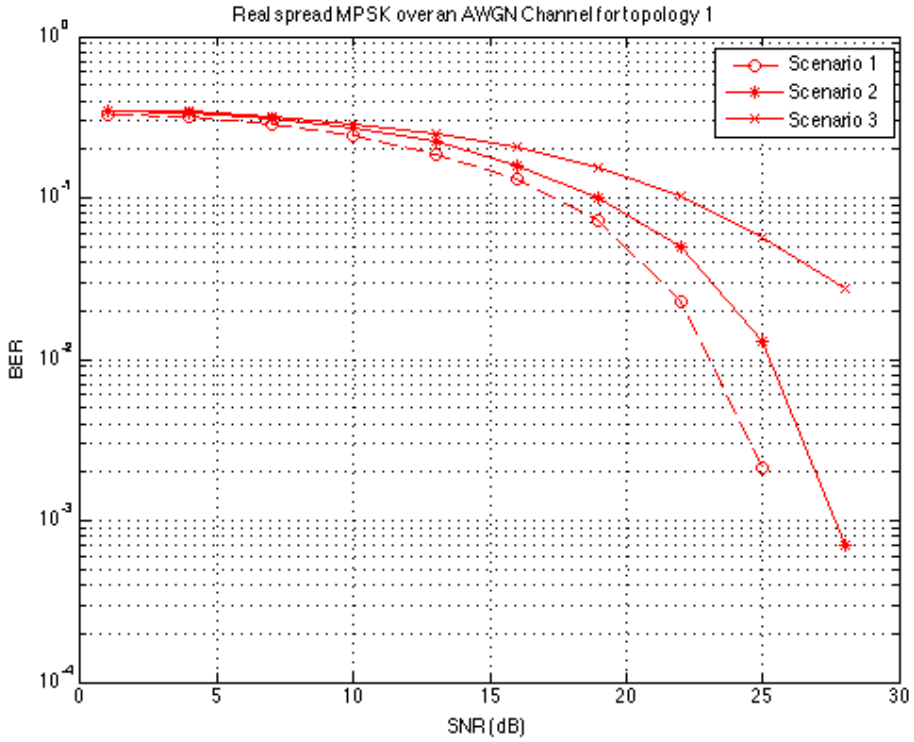

Figure 5.11. BER Performance of DS-BPSK for scenario-1

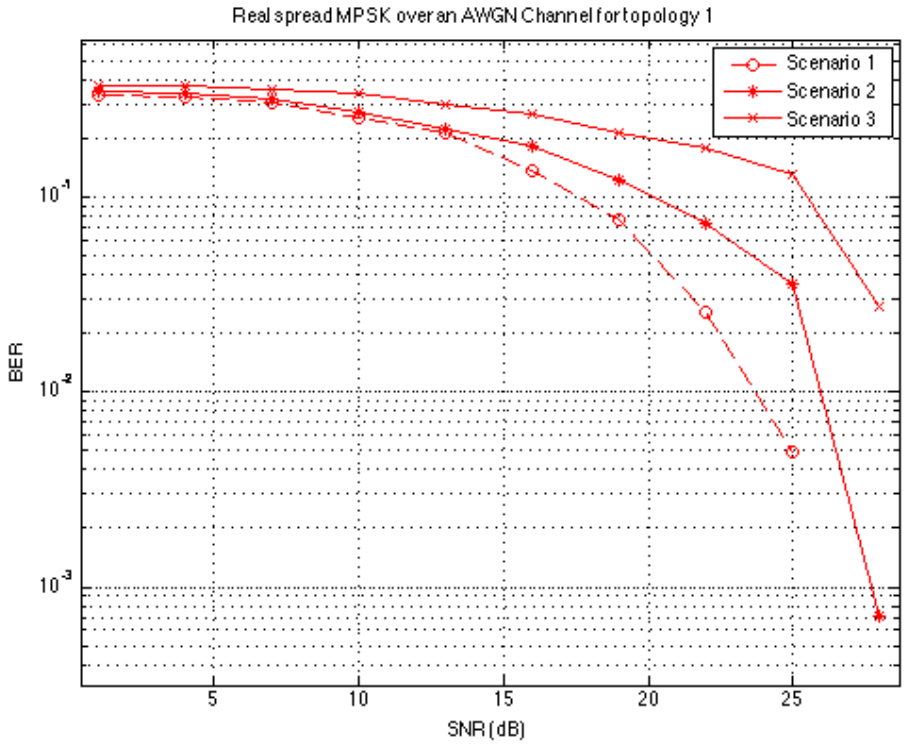

Figure 5.12. BER Performance of DS-BPSK for scenario-2

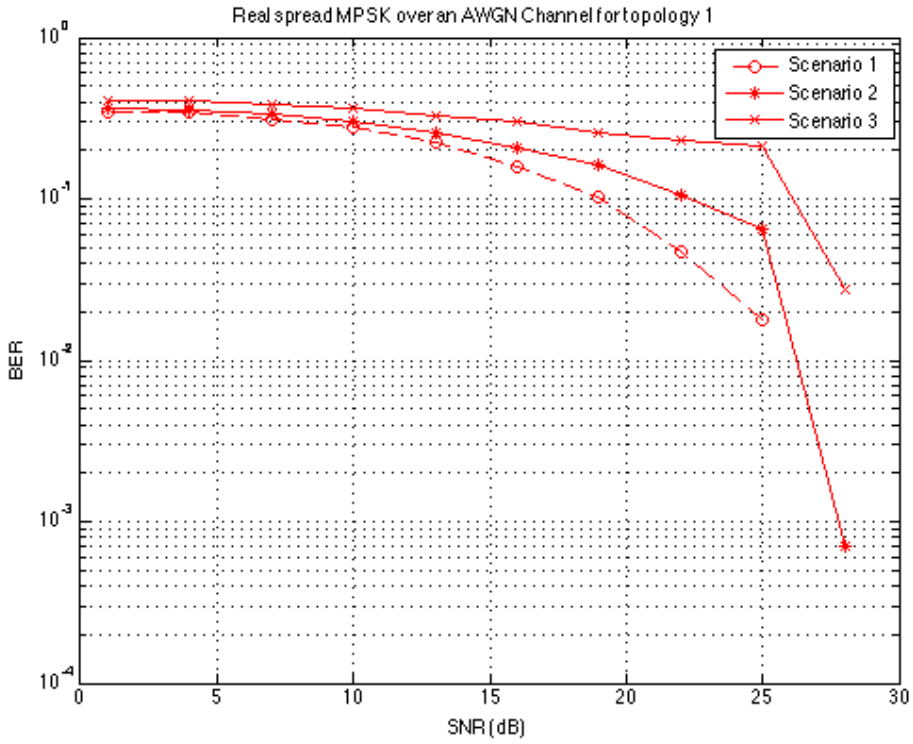

Figure 5.13. BER Performance of DS-BPSK for scenario-3

6. DISCUSSION OF EXPERIMENTAL EVALUATION

In this section, an experimental evaluation of the proposed scheme is presented using the amplify-and-forward technique. Performance of the proposed scheme is evaluated using two modulation techniques: DS-DBPSK and DS-OFDM. Additionally, the experimental results are compared with the simulations. A step-by-step procedure of the experimentation is explained in this section.

The experimental setup includes three USRP2s. They are connected via a 1Gbit switch to a computer running Ubuntu 12.04. The gnuradio-companion software tool is used to interface with and control the USRP2s. The WBX daughterboard is installed in all USRPs, which enabled usage of a carrier frequency around 2.1 GHz. The three nodes are initially placed in an equilateral triangle topology. The distance between each of the nodes is equal to 50 inches or 1.27m as shown in Fig. 6.1. The topology dimensions and corresponding transmission powers are scaled down in order to fit inside the laboratory.

Figure 6.1. Experimental Setup

The proposed multi-variable optimization algorithm is implemented for both DBPSK and OFDM-BPSK and the BER performance is compared. Additionally, we compared performance of the amplify-and-forward and decode-and-forward techniques using the DBPSK implementation. We observe massive degradations in amplify and forward as compared to the decode and forward method. The sender, relay, and destination node are setup as shown in the Fig. 6.1, while the control and monitoring processing is performed by dedicated computer.

The experimentation process is executed sequentially. In timeslot one, we generate a random sequence of bits at a sample rate 100k, for both the DBPSK and OFDM implementation. In the case of DBPSK, we generate a random bit sequence and then spread it by PN sequence of length $(2^(11) - 1)$. This spreading is shown in the Fig. 6.2.

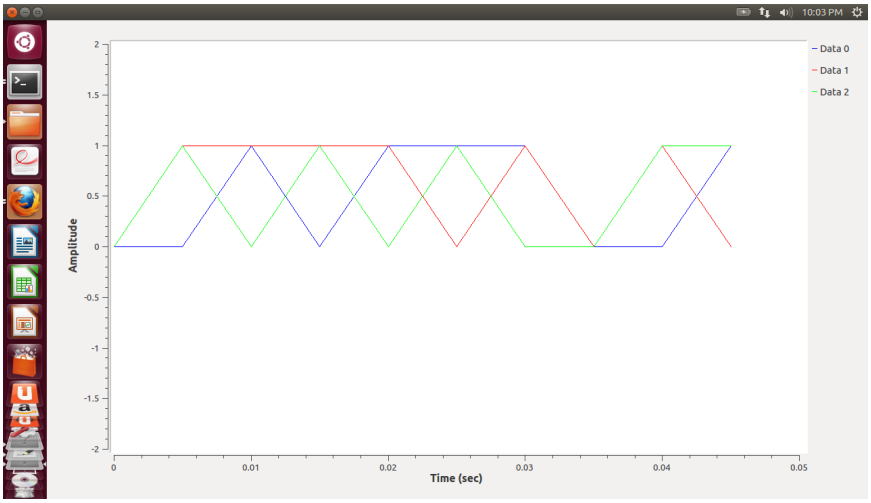

Figure 6.2. Spreading of a Single Input Bit by a 10-bit PN Sequence

In the Fig. 6.2, "data 1" is spread using the PN sequence and gives out the "data 3". Spreading increases the resistance to background noise level, makes it possible to share frequency with other signals and helps in maintaining proper signal to noise ratio. This is then modulated to DBPSK symbols and also the individual spread subcarriers are converted to OFDM symbols.

The modulated symbols are transmitted to the relay node and the destination node. At the relay node and the destination node, the received symbols are stored in the their respective *.dat* files for further offline processing. Note, that in final implementation the forwarding and processing will be performed in-line. The power spectrum of the transmitted and received DBPSK signal is shown in Fig. 6.3.

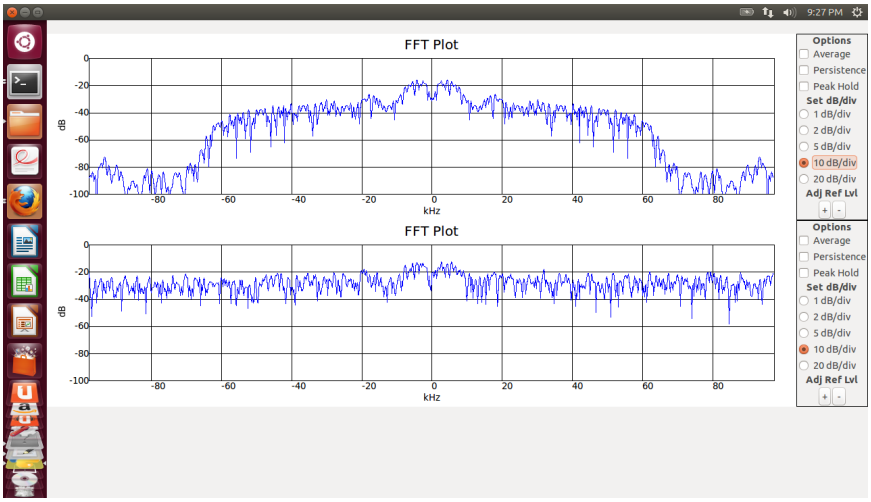

Figure 6.3. Power Spectrum of Transmitted and Received DBPSK Modulated Signal

We employ a high speed PN sequence along with the random generated bits to modulate the carrier. The resulting DBPSK signal is centered at carrier frequency of 2.1 GHz with the main lobe bandwidth equal to the twice the rate of the modulation code (from null to null). In the shown results, it is bandwidth from -10Khz to 10Khz that contains the bell shaped curve with a peak at 0Khz. The first harmonic is observed in the bandwidth from 10 to 20Khz with a peak at 15Khz. Similarly, this continues until 70Khz. Making the total bandwidth as 140khz. The figure shows that the bell shaped curve decreases in amplitude when the harmonics weaken.

The Fig. 6.4 shows the power spectrum of the transmitted OFDM Signal with FFT length as 512, occupied tones as 200 and the cyclic prefix as 128. Here, the OFDM signal spectrum is corresponding to a spread random input binary sequence. The 0 (DC) input is set to zero in the IFFT implementation. This is clearly observed

in the frequency spectrum at the zero on the graph. The subcarriers at the edges are kept unused to allow for a reasonable transition band at the channel boundaries. As we use OFDM over Spread BPSK modulation, each subcarrier carries 1 bit of information, which is further spread by the PN chip sequence.

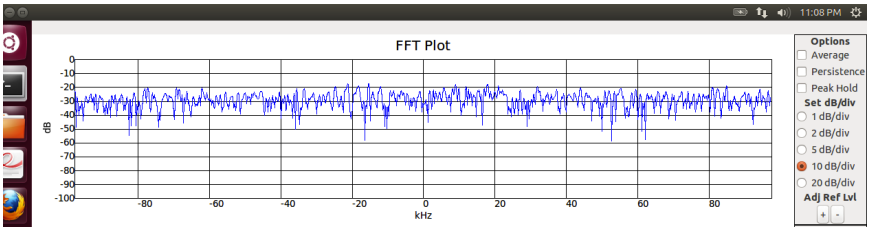

Figure 6.4. Power Spectrum of the transmitted OFDM Signal with FFT length = 512 and Occupied Tones = 200

The major advantage of using GNU Radio companion is that all the necessary useful information can be saved in .dat files. They can be further processed in Octave using Matlab scripts. We use two forms of co-operative relay schemes to test our multivariable optimization algorithm. In the Amplify and Forward Scheme, we receive the signal at the relay during time slot 1 and then amplify and propagate it to the final destination. Henceforth, we amplify both the noise and the signal at the same time. This gives a high bit-error rate. Hence, the sequence of steps in Amplify and Forward Method are, in time slot 1 we generate the signal, spread it a PN sequence intended for the relay node. We add this to an equally spread signal intended for the destination node.

This additive signal is modulated at a sampling rate of 195.312Khz and a carrier frequency of 2.1Ghz and transmitted to both the relay and destination node. This is further stored in files *relay_complex.dat* and *dest1_complex.dat* respectively. The entire GNU Radio Companion code for time slot 1 is included in the file *slot_one.grc*. In the second time slot, the relay node re-amplifies and retransmites the stored data, as implmented in the script file *slot_two.grc*. In case of the decode-and-forward technique, the signal is decoded at the relay in the file *phase1_decode.grc* and then re-transmitted.

This significantly reduces the bit error rate when compared with the amplify-and-forward technique since the decoding removes the analog noise before forwarding. The decoded, digital data contain some bit-errors but the correctly decoded bits are retransmitted without the distortions and noise. In contrast, there are amplified in the amplify-and-forward techniques thus increasing the received noise and distortion at the destination node.

In case of amplify-and-forward technique, the noise appended at the relay received signal is assumed to be additive white gaussian noise with zero mean and variance N_0. Although higher modulation schemes transmit the m (no of bits per symbol) times the amout of information in the same time and bandwidth, the disadvantage is that the BER increases with increase in number of bits per symbol. We use Differential BPSK as our underlying modulation scheme because as the higher the amount of phase changes, the higher is the probability of error. This is because the received constellation points get closer and decoding errors arise.

The main reason behind using OFDM like technique is that it provides resistance to narrow-band and impulsive noise and also reduces channel equalizer complexity. This combined with spreading greatly improves robustness in frequency selective channels, provided the symbol rate is low. In the case of high symbol rates, spreading causes degradation in channel performance due to high interference.

We compare the DSSS-BPSK Scheme experimental values of BER vs SNR with the three simulated DSSS-BPSK values. These three simulated plots represent the same scheme implemented using Simulink with different multipath delays in the previous section. The signal-to-noise ratio of the receiving USRP2 nodes are adjusted in such a manner so as to adjust the link power according to our scheme. We use a transmit power such that link 1-2 receives 15dbm, link 1-3 receives 16.5dbm and link 3-2 receives the maximum of 18.45dbm. This is the maximum supported power of USRP2 with WBX daughterboard with a maximum frequency of 2.2GHz. For spreading of the information, we use a PN code of length 2^{11-1}.

The SNR vs BER of DSSS-BPSK is compared with the simulated BER of BPSK Modulated signal for our multi-variable optimization scheme. Both the two types of cooperative schemes are used: amplify-and-forward and decode-and-forward. As we can see, the performance of decode and forward Fig. 6.5 is better than amplify and

forward scheme in many places. We can observe that the amplify-and-decode scheme performs better for SNR from source to relay greater than a threshold value (15dbm in this case). Over the threshold value of 15dbm, amplify and decode scheme almost follows the simulated BER curve. But, below 15dbm of SNR ratio, its performance is way worse than all the multipath simulated schemes. This is because of the noise getting appended to the received signals at the relay.

We observe that, the decode and forward scheme as in Fig. 6.5 performs well enough in both the lower SNR and high SNR conditions. It follows the simulated BER curve for all the multipath simulated scenarios. On the contrary, we can see that even though decode and forward scheme performs way better than amplify and forward scheme for our defined optimized power values, it still does not BER equivalent to the zero multipath simulated case.

We particularly compare our scheme to the zero multipath case because in our experimental scenario, we do not place any obstacles and also place our nodes in the center of the lab, hence avoiding as many multipath echoes from the walls as possible. Fig. 6.6 shows the Bit error rate vs SNR ratio for the setup we considered with each node separated by 50 inches from each other. It can be observed that the performance of amplify and forward scheme is worse than the decode-and-forward scheme. This is because of the noise, which gets amplified at the relay node in the amplify-and-forward scheme.

Of the three simulated curves that we observe, the one with the maximum bit error rate with lowest possible SNR ratio is the simulated path with zero multipath error. In our case, this corresponds to iteration 1 in the simulation scenario. This was clearly explained in the previous results section. You can also observe in the experimental scenario that, both the amplify and forward scheme as well as the decode and forward try to follow the simulated curves until 30db SNR, at which point all three of the simulated curves reach their minimum BER limit. After 30db SNR, the experimental BER still continues to achieve the simulated BER limit and ends up performing a little less well than the simulated BER limit. Of all the curves, the experimental curve best follows the simulation curve iteration 2. This iteration 2 reflects a multipath scenario of delay 2 and a minimum symbol rate.

Fig. 6.7 shows the comparison of BER vs SNR for Experimental DSSS-OFDM (BPSK) with Experimental DSSS-DBPSK for our multi-variable optimization scheme. Once having implemented our optimization scheme with DSSS-BPSK for topology, we get a bit error rate of 1.9 % for amplify-and-forward scheme and a bit error rate of 1.7 % for decode and forward scheme. We implement the same scheme using DSSS-OFDM as shown in Fig. 6.8. The reason for implementing the same scheme using OFDM is that, the combination of orthogonal carriers and spreading helps in eliminating inter-cell interference in a cellular environment. Also, the use of cyclic prefix helps in eliminating inter-symbol interference.

The performance of DSSS system depends upon the space available, power used and the complexity of the receiver used. Usually DSSS systems use power efficient PSK modulation, that is, the probability of error for a given SNR for PSK systems is less. Also, the higher the chip rate of PN code, the smaller the degradation to multipath for a DSSS system. Using all of this as a basis, we implement our scheme for the second time in DSSS-OFDM systems with BPSK as the underlying modulation scheme for each of the sub-carriers. Although many of the papers before implement OFDM with QAM modulation, we implement it using BPSK because we use it as a comparison means with the previous DSSS-BPSK system.

Similar to DSSS-BPSK, for spreading of the information, we use a PN code of length 2^{11-1}. The co-operative scheme that we use for DSSS-OFDM implementation is decode and forward method, because, with amplify and forward method, the error during the combining of direct and relay signals at the destination is too high and degradation due to multipath becomes too high. So, we implement our scheme only using decode and forward method. We observe that the BER of OFDM systems using BPSK scheme outperforms both the amplify-and-decode scheme of DSSS-BPSK and decode-and-forward scheme of DSSS-BPSK.

The experimental results of the DSSS-OFDM with BPSK indicate that it is more suitable for small distance communication. We see, that DSSS-OFDM performs at a BER of 1.06 % for our topology, which is just a few point decimals below our DSSS-BPSK scheme. We can observe that spreading makes the system able to perform at higher SNR values.

Meaning that, we can extend this scheme to be able to handle OFDM over higher-order PSK modulations without the equivalent degradations in BER for longer distances. Hence, we can recommend our scheme for sub-optimal power allocation with DSSS-OFDM (BPSK) for smaller distance communication.

6.1. OBSERVATIONS FROM THE EXPERIMENTAL RESULTS

We implement our optimal resource allocation scheme using direct spreading over power efficient PSK modulation scheme. The probability of error PSK modulation is quite low. By using direct spreading technique, which is self-synchronizing, together with the DBPSK and OFDM, we maximize the performance in terms of BER. The experiments show that the proposed power allocation minimizes the BER for shorter distances using a lower-order (2=binary) PSK modulation. We use a spreading scheme with 2^{11-1} bit PN code in-order to improve the performance of our system over longer distances. The scheme supports longer distances by using OFDM with a higher order PSK scheme over spreading. Hence by comparing the BER performance for our scheme, we see that, we achieve a BER of 1.06 % for the DS-OFDM-BPSK, a BER of 1.7 % for DS-DBPSK decode and forward scheme and a BER of 1.9 % for DS-DBPSK amplify and forward scheme. We conclude that we would prefer DS-OFDM-BPSK over both the DS-DBPSK co-operative schemes for short and long distance communication.

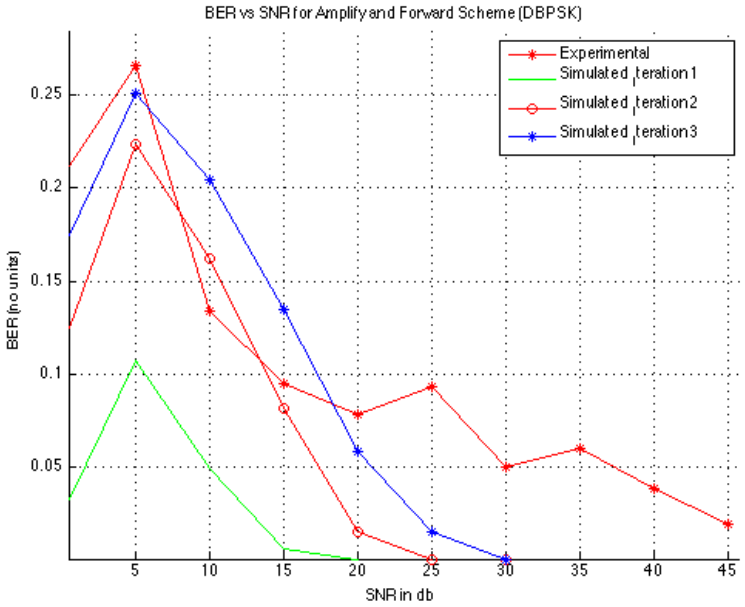

Figure 6.5. SNR vs BER Plot for DSSS-BPSK Modulation (Amplify and Forward Scheme)

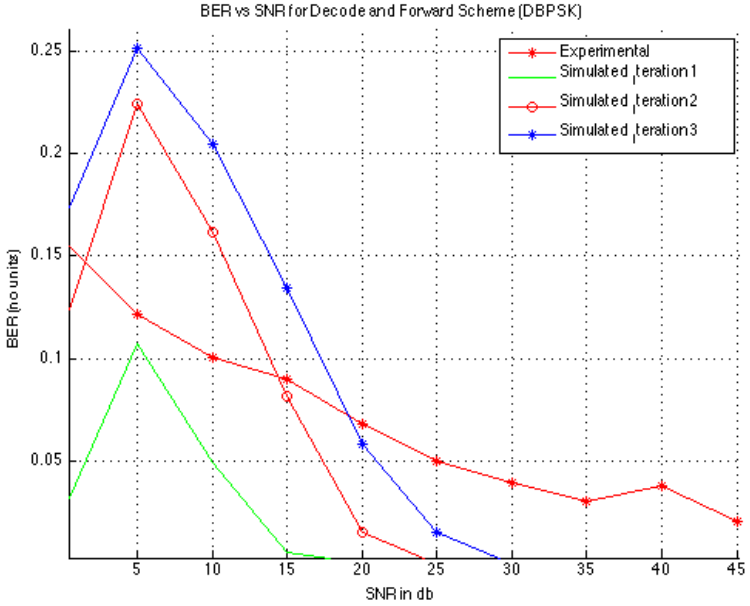

Figure 6.6. SNR vs BER Plot for DSSS-BPSK Modulation (Decode and Forward Scheme)

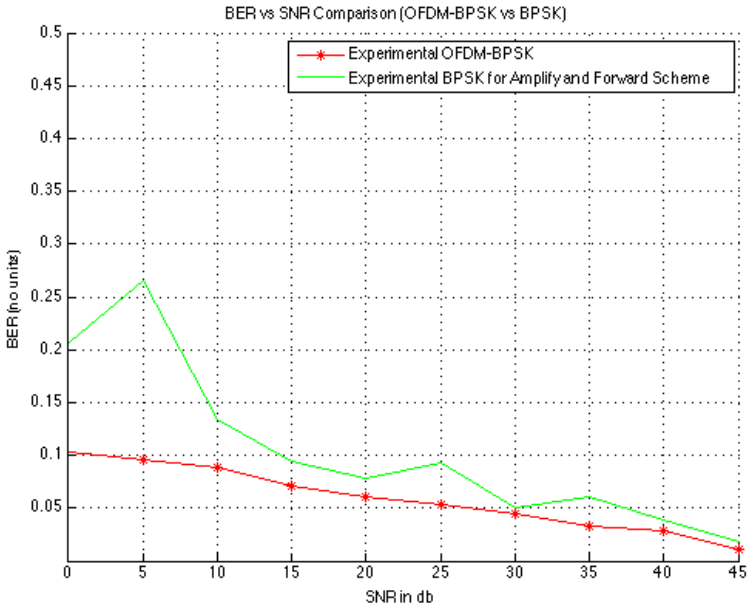

Figure 6.7. SNR vs BER Plot Comparison between DSSS-BPSK Modulation and DSSS-OFDM Modulation (Amplify and Forward Scheme)

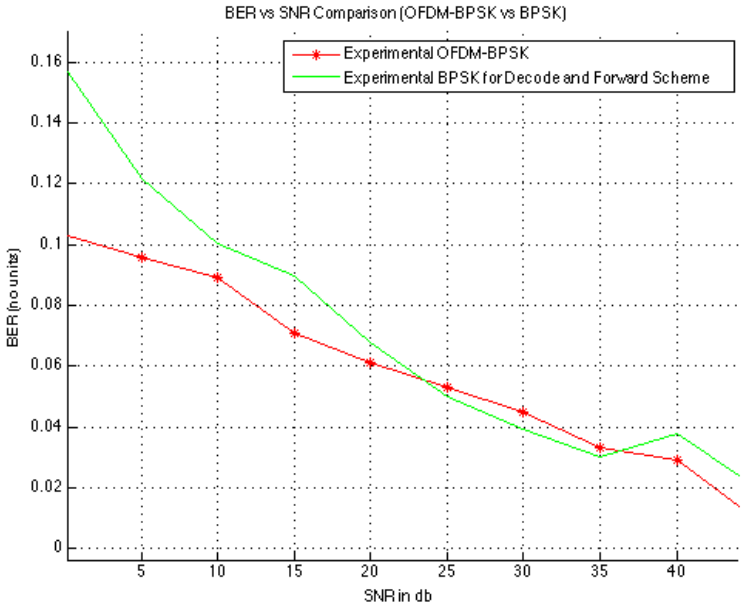

Figure 6.8. SNR vs BER Plot Comparison between DSSS-BPSK Modulation and DSSS-OFDM Modulation (Decode and Forward Scheme)

7. CONCLUSIONS AND FUTURE WORK

In the first section, the issue of fair and sub-optimal resource allocation in wireless networks is dealt with by taking up power allocation as the main issue. A scheme for optimal power allocation is developed for a particular three-node topology and is extended to further 3-more nodes. Then the scheme is simulated with Matlab using realistic assumptions as to fading criteria, shadowing criteria and path loss effects. A particular set of power values are deemed to the most optimal of the set of values. These optimal power values for a given scenario are estimated. Then the solution is further tested in Simulink based on DSSS-DBPSK modulation scheme. With the sub-optimal values keyed in, the Simulink model further tests the prototype with multipath fading conditions and the bit error rate for each of these iterations is plotted. The BER takes a value in the range of 10^{-2} to 10^{-3}, which is sub-optimal range for co-operative relay scheme that we implement. Hence, theoretical analysis indicate that, in a network, for every 3-hops, the optimal resource can be allocated by setting the power of the relay to destination link to the maximum power threshold and the other two links set according to the power allocation scheme. We conclude by pointing out that the simulation results demonstrate efficiency of the scheme proposed and also indicate the trade-off between the throughput, bit-error rate and the fairness involved.

The second section involves implementing the sub-optimal performance maximization scheme in real time using software-defined radios (USRP2) with the help of GNU Radio framework. The implementation is only carried out for the first three hops and then the bit error rate is observed for DSSS-DBPSK and DSSS-OFDM using cooperative relay schemes. We observe that our scheme performs well enough for DSSS-DBPSK scheme and follows the bit error rate curve for the decode-and-forward scheme. Also, since we use a PN sequence of length 2^{11-1} for spreading the data, our scheme works for short distance and long distance communication. We see that our scheme performs best with DSSS-OFDM-BPSK modulation. In conclusion, a total BER of 1.06 % for the OFDM scheme offers a much end-to-end performance efficiency as compared to many real-time implementations.

Future work would include testing the proposed scheme in a larger network with more complex scenarios. This would also include implementing this particular scheme using OFDM-QAM and CDMA scheme. Also, future work should include evaluation of the scheme in more realistic scenarios with a set of obstacles and mobile interference sources.

BIBLIOGRAPHY

[1] D. Bertsekas and R. G. Gallager, *Data Networks. Prentice Hall, 1991.*

[2] S. Boyd and L. Vandenberghe, *Convex Optimization. Cambridge University Press, 2004.*

[3] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability", *Journal of the Operational Research Society, pp. 237-252, 1998.*

[4] L. Butty'an and J. P. Hubaux, "Report on a Working Session on Security in Wireless Sensor Networks", *ACM SIGMOBILE Mobile Computing and Communications Review, 2002, pp. 74-94.*

[5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, K. S. J. Pister, "System Architecture directions for networked sensors", in: *Proceedings of the Ninth International Conference on Architectural Support for Programming languages and Operating Systems, 2000, pp. 93-104.*

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", *IEEE communications magazine, vol. 40, num. 8, 2002, pp. 102-114.*

[7] S. Roundy, P. K. Wright and J. Rabaey, "A Study of low-level vibrations as a power source for wireless sensor networks", *Computer Communications, Volume 26, Issue 11, 1 July 2003, pp. 1131-1144.*

[8] B. W. Wah and Z. Wu, "The theory of discrete Lagrange multipliers for nonlinear discrete optimization", *Principles and practice of Constraint Programming, October 1999, pp. 28-42.*

[9] R. Canchi and Y. Akaiwa,"Performance of adaptive transmit power control in p/4 DQPSK mobile radio systems in flat Rayleigh fading channels", *Proceedings of the IEEE Vehicular Technology Conference, Vol. 2, 1999, pp.1261-1265.*

[10] H. W. Tucker and A. W.,"Nonlinear Programming", *Proceedings of 2nd Berkeley Symposium, Berkeley: University of California Press, 1951, pp. 481-492.*

[11] D. Li, B. Q. Wang, L. Guan, Q. Liu,"Analysis for Market-driven Multipath Inter-domain Routing ", *Proceedings of Third International Symposium on Intelligent Information Technology Application, vol. 2, 2009, pp.360-363.*

[12] E. B. Rodrigues, F. Casadevall,"Control of the trade-off between resource efficiency and user fairness in wireless networks using utility-based adaptive resource allocation ", *IEEE Communications Magazine, vol.49, Issue.9, 2011, pp.90-98.*

[13] X. Cao, J. Chen, Y. Sun, X. Shen,"Maximum Throughput of IEEE 802.15.4 Enabled Wireless Sensor Networks ", *Proceedings of IEEE Global Telecommunications Conference, 2010, pp.1-5.*

[14] M. Tekaya, N. Tabbane, S. Tabbane,"Multipath routing mechanism with load balancing in ad hoc network ", *Proceedings of International Conference on Computer Engineering and Systems, 2010, pp.67-72.*

[15] Y. Fang and L. Wang,"An Algorithm of Static Load Balance Based on Topology for MPLS Traffic Engineering", *Proceedings of International Conference on Information technology (ICIE), vol.2, 2009, pp.26-28.*

[16] J. Crichigno,M. Y. Wuz, S. K. Jayaweeray and W. Shuy,"Throughput Optimization in Multi-hop Wireless Networks with Multi-packet Reception and Directional Antennas", *Parallel and Distributed Systems, IEEE Transactions on, vol.22, Issue.7, July 2011, pp.1206-1213.*

[17] P. Gupta and P. R. Kumar,"The capacity of wireless networks", *Information Theory, IEEE Transactions on, Vol.46, Issue.2, 388404, Mar 2000.*

[18] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu," Impact of interference on multi-hop wireless network performance", *In Mobicom 2003, San Diego, CA, 2003.*

[19] M. Kodialam and T. Nandagopal," Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks", *In ACM Mobicom 2005, New York, NY, pp. 73-87, 2005.*

[20] P. Kyasanur and N. Vaidya, Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces, *In ACM Mobicom 2005, New York, NY, pp. 43-57, 2005.*

[21] M. J. Neely, Energy Optimal Control for Time Varying Wireless Networks", *IEEE Transactions on Infomation Theory, vol. 52, no. 2, pp.2915-2934, July 2006.*

[22] J. Kazemitabar, H. Yousefizadeh, and H. Jafarkhani, Impact of Physical Layer Parameters on Connectivity of Ad Hoc Networks", *Proceedings of IEEE ICC 2006, vol.4, pp.1891-1896, june 2006.*

[23] L. Lin, X. Lin and N. Shroff, Low-Complexity and Distributed Energy Minimization in Multi-hop Wireless Networks", *Networking, IEEE/ACM Transactions on, vol.18, Issue.2, pp. 501-514, april 2010.*

[24] M. Johansson and L. Xiao,"Cross-layer optimization of wireless networks using nonlinear column generation", *Wireless Communications, IEEE Transactions on, vol.5, Issue.2, pp.435-445, february 2006.*

[25] R. L. Cruz and A. Santhanam,"Optimal routing, link scheduling and power control in multi-hop wireless networks", *In Proceedings of the 2003 Infocom,San Francisco, CA, April 2003.*

[26] G. Youngblood, AC5OG, A Software Defined Radio for the Masses, Part 1, *QEX, Jul/Aug 2002, pp 13-21.*

[27] G. Youngblood, AC5OG, A Software Defined Radio for the Masses, Part 2", *QEX, Sep/Oct 2002, pp 10-18.*

[28] P. Burns," Software Defined Radio for 3G", *Artech House Inc., 2002.*

[29] W. Tuttlebee, "The Software Defined Radio: Enabling Technologies", *John Wiley and Sons, 2002.*

[30] GNU Radio WikiStart (2013), "Welcome to GNU Radio", `Available:http://www.gnuradio.org/redmine/projects/gnuradio/wiki/WikiStart`.

[31] E. Blossom, "GNU Radio: Tools for Exploring the RF Spectrum", *Linux Journal, Issue 22, June 2004.*

[32] J. Blum (2009), "GNU Radio Companion", `Available:http://www.joshknows.com/grc`.

[33] G. Van Rossum , "An Introduction to Python for UNIX/C Programmers", *Proceedings NLUUG najaarsconferentie (Dutch UNIX users group, 1993.*

[34] B. Stroustrup, "The C++ Programming Language (Third Edition)", *Addison Wesley Professional, June 30, 1997.*

[35] Wikipedia (2013, Jan), " Universal Software Radio Peripheral", `Available:http://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral`

[36] Ettus Research LLC , ed. ,"USRP2: The Next Generation of Software Radio Systems", *Mountain View, CA, USA.*

[37] A. Csete (Jan 2010), " The WBX 50-2200 MHz transceiver", `http://www.oz9aec.net/index.php/gnu-radio/gnu-radio-blog/301-the-wbx-50-2200-mhz-transceiver`.

[38] *www.gnuradio.org.*

[39] P. Balister and J. H. Reed,"USRP Hardware and Software Description", *MPRG/Wireless@Virginia Tech, June 30, 2006.*

[40] T. M. Schmidl and D. C. Cox,"Robust frequency and timing synchronization for OFDM",*IEEE Trans. on Communications, pp-1613-1621, 1997.*

[41] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behaviour", *IEEE Transactions on Information Theory, Volume 50, No. 12, Dec 2004.*

[42] "GNU Radio Blocks", March 2013 `http://gnuradio.org/redmine/projects/gnuradio/repository/revisions/master/show/gr-digital`

[43] D. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Transactions on Communications, Vol. 28, No. 11, pp. 1867 - 1875, 1980."*

[44] G. R. Danesfahani and T. G. Jeans, "Optimisation of modified Mueller and Muller algorithm,"*Electronics Letters, Vol. 31, no. 13, 22 June 1995, pp. 1032 - 1033."*

VITA

Vinodhini Ravikumar obtained her Bachelors in Electronics and Communication Engineering from Anna University, Chennai, India in 2010. During her senior year, she interned at the Indian Satellite Research Organization, Bangalore, India where her project was on developing a Micro-strip patch antenna for GPS and radar systems. In the Fall 2010, she started her graduate studies at Missouri University of Science and Technology, Rolla MO, USA with a concentration on wireless sensor networks and communications. During this time she worked as a Graduate Research Assistant with the Embedded Control Systems and Networking Lab at the Dept. of ECE, Missouri S&T. She received her Master's degree in Electrical Engineering from Missouri University of Science and Technology in May 2014.